Assignment 1 Solutions

MapReduce and PageRank

Question 1:

Suppose our input data to a map-reduce operation consists of integer values (the keys are not important). The map function takes an integer i and produces the list of pairs (p,i) such that p is a prime divisor of i. For example, map (12) = [(2,12), (3,12)]. The reduce function is addition. That is, reduce (p, [i_1, i_2, ..., i_k]) is (p,i_1+i_2+...+i_k). Compute the output, if the input is the set of integers 15, 21, 24, 30, 49.

Answer 1: The output of map function is

map (15) = [(3, 15), (5, 15)]

map (21) = [(3, 21), (7, 21)]

map (24) = [(2, 24), (3, 24)]

map (30) = [(2, 30), (3, 30), (5, 30)]

map (49) = [(7, 49)]

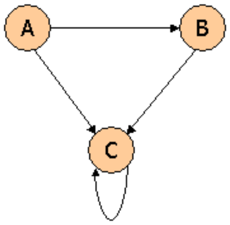
These are the respective prime divisors of inputs

The output of reduce function is

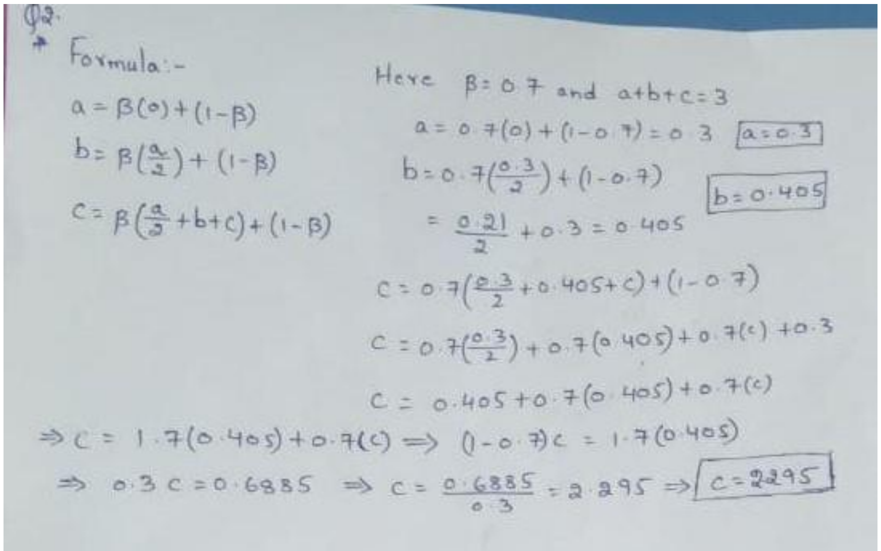
reduce (2, 54), reduce (3,90), reduce (5,45), reduce (7, 70).

Question 2:

Consider three Web pages with the following links:



Suppose we compute PageRank with a β of 0.7, and we introduce the additional constraint that the sum of the Page Ranks of the three pages must be 3, to handle the problem that otherwise any multiple of a solution will also be a solution. Compute the Page Ranks a, b, and c of the three pages A, B, and C, respectively.



Q2.

* Formula:-

$a = \beta(0) + (1-\beta)$

$b = \beta\left(\frac{a}{2}\right) + (1-\beta)$

$c = \beta\left(\frac{a}{2} + b + c\right) + (1-\beta)$

Here $\beta = 0.7$ and $a+b+c=3$

$a = 0.7(0) + (1-0.7) = 0.3$ $\boxed{a=0.3}$

$b = 0.7\left(\frac{0.3}{2}\right) + (1-0.7)$ $\boxed{b=0.405}$

$= \frac{0.21}{2} + 0.3 = 0.405$

$c = 0.7\left(\frac{0.3}{2} + 0.405 + c\right) + (1-0.7)$

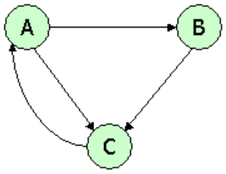
$c = 0.7\left(\frac{0.3}{2}\right) + 0.7(0.405) + 0.7(c) + 0.3$

$c = 0.405 + 0.7(0.405) + 0.7(c)$

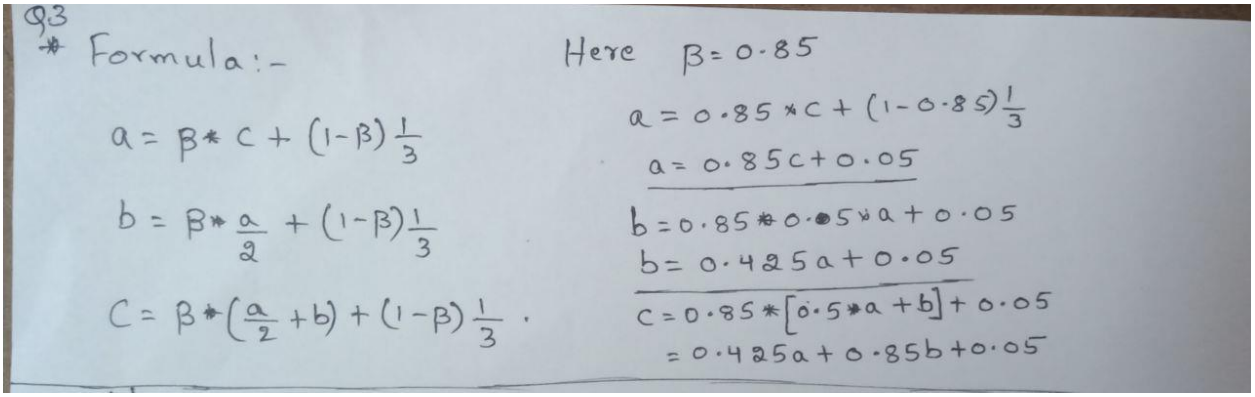
$\Rightarrow c = 1.7(0.405) + 0.7(c) \Rightarrow (1-0.7)c = 1.7(0.405)$

$\Rightarrow 0.3c = 0.6885 \Rightarrow c = \frac{0.6885}{0.3} = 2.295 \Rightarrow \boxed{c=2295}$

Question 3:



Suppose we compute PageRank with β=0.85. Write the equations for the Page Ranks *a*, *b*, and *c* of the three pages A, B, and C, respectively.



Q3

* Formula:-

$a = \beta * c + (1-\beta)\frac{1}{3}$

$b = \beta * \frac{a}{2} + (1-\beta)\frac{1}{3}$

$c = \beta * \left(\frac{a}{2} + b\right) + (1-\beta)\frac{1}{3}$.

Here $\beta = 0.85$

$a = 0.85 * c + (1-0.85)\frac{1}{3}$

$\underline{a = 0.85c + 0.05}$

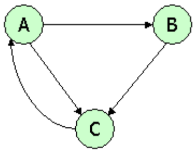
$b = 0.85 * 0.5 * a + 0.05$

$\underline{b = 0.425a + 0.05}$

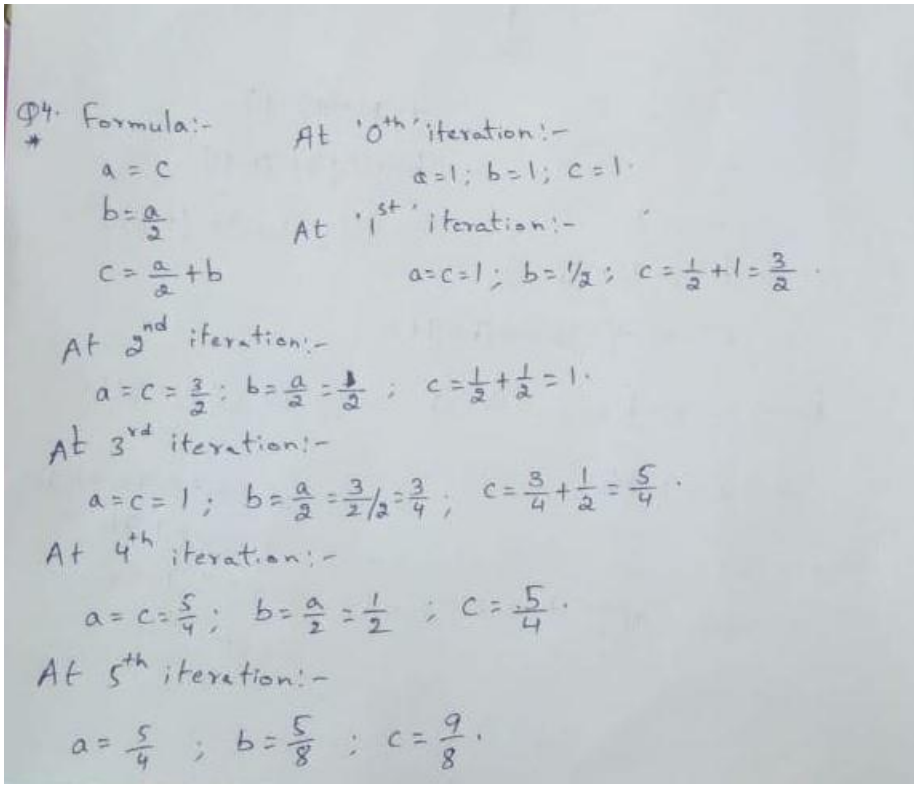
$c = 0.85 * [0.5 * a + b] + 0.05$

$= 0.425a + 0.85b + 0.05$

Question 4:



Assuming no "taxation," compute the Page Ranks *a*, *b*, and *c* of the three pages A, B, and C, using iteration, starting with the "0th" iteration where all three pages have rank *a* = *b* = *c* = 1. Compute as far as the 5th iteration, and also determine what the Page Ranks are in the limit.



Q4. Formula:-

*

$a = c$

$b = \frac{a}{2}$

$c = \frac{a}{2} + b$

At '0th' iteration:-

$a = 1;\ b = 1;\ c = 1.$

At '1st' iteration:-

$a = c = 1;\ b = 1/2;\ c = \frac{1}{2} + 1 = \frac{3}{2}.$

At 2nd iteration:-

$a = c = \frac{3}{2};\ b = \frac{a}{2} = \frac{1}{2};\ c = \frac{1}{2} + \frac{1}{2} = 1.$

At 3rd iteration:-

$a = c = 1;\ b = \frac{a}{2} = \frac{3}{2}/2 = \frac{3}{4};\ c = \frac{3}{4} + \frac{1}{2} = \frac{5}{4}.$

At 4th iteration:-

$a = c = \frac{5}{4};\ b = \frac{a}{2} = \frac{1}{2};\ c = \frac{5}{4}.$

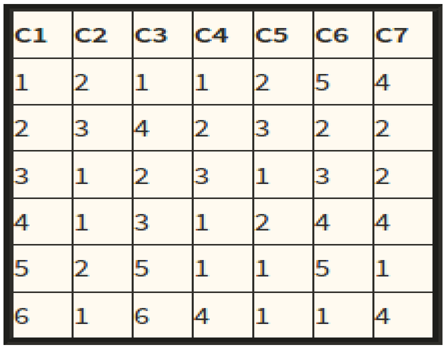
At 5th iteration:-

$a = \frac{5}{4};\ b = \frac{5}{8};\ c = \frac{9}{8}.$

Locality-Sensitive Hashing

Question 1:

Here is a matrix representing the signatures of seven columns, C1 through C7.



C1	C2	C3	C4	C5	C6	C7
1	2	1	1	2	5	4
2	3	4	2	3	2	2
3	1	2	3	1	3	2
4	1	3	1	2	4	4
5	2	5	1	1	5	1
6	1	6	4	1	1	4

Suppose we use locality-sensitive hashing with three bands of two rows each. Assume there are enough buckets available that the hash function for each band can be the identity function (i.e., columns hash to the same bucket if and only if they are identical in the band). Find all the candidate pairs.

Answer:

The candidate pairs are:
In Band1 → C1 and C4, C2 and C5
In Band2 → C1 and C6
In Band3 → C4 and C7, C1 and C3

Question 2:

Suppose we have computed signatures for a number of columns, and each signature consists of 24 integers, arranged as a column of 24 rows. There are N pairs of signatures that are 50% similar (i.e., they agree in half of the rows). There are M pairs that are 20% similar, and all other pairs (an unknown number) are 0% similar.
We can try to find 50%-similar pairs by using Locality-Sensitive Hashing (LSH), and we can do so by choosing bands of 1, 2, 3, 4, 6, 8, 12, or 24 rows. Calculate approximately, in terms of N and M, the number of false positive and the number of false negatives, for each choice for the number of rows. Then, suppose that we assign equal cost to false positives and false negatives (an atypical assumption). Which number of rows would you choose if M: N were in each of the following ratios: 1:1, 10:1, 100:1, and 1000:1?
Answer:
If M: N = 1: 1, then the number of rows = 3
If M: N = 10: 1, then the number of rows = 2
If M: N = 100: 1, then the number of rows = 1
If M: N = 1000: 1, then the number of rows = 1

Question 3:

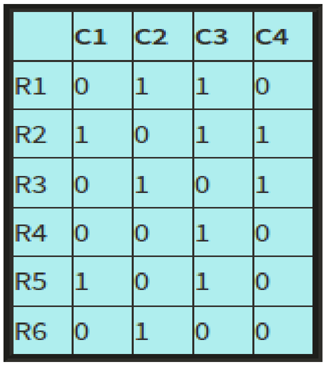
Find the set of 2-shingles for the "document":
ABRACADABRA
and also, for the "document":
BRICABRAC
Answer:
Set of 2 shingles for ABRACADABRA **is [(AB,2), (BR,2), (RA,2), (AC,1), (CA,1), (AD,1), (DA,1)]**
Set of 2 shingles for BRICABRAC **is [(BR,2), (RI,1), (IC,1), (CA,1), (AB,1), (RA,1), (AC,1)]**
Answer the following questions:

1. How many 2-shingles does ABRACADABRA have?
   Seven (7)
2. How many 2-shingles does BRICABRAC have?
   Seven (7)
3. How many 2-shingles do they have in common?
   Five (5)
4. What is the Jaccard similarity between the two documents"?

   Jaccard similarity = sets intersection / sets union = 5/9.

Question 4:

Consider the following matrix:



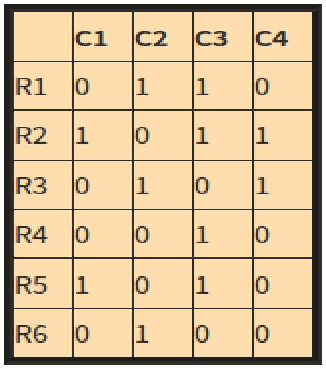
	C1	C2	C3	C4
R1	0	1	1	0
R2	1	0	1	1
R3	0	1	0	1
R4	0	0	1	0
R5	1	0	1	0
R6	0	1	0	0

Compute the Jaccard similarity between each pair of columns.

Answer:
Jaccard similarity of C1 and C2 is SIM (C1, C2) = 0/5
Jaccard similarity of C2 and C3 is SIM (C2, C3) = 1/6
Jaccard similarity of C3 and C4 is SIM (C3, C4) = 1/5
Jaccard similarity of C4 and C1 is SIM (C4, C1) = 1/3

Jaccard Similarity of C1 and C3 is SIM (C1, C3) = 2/4
Jaccard Similarity of C2 and C4 is SIM (C2, C4) = 1/4

Question 5: Consider the following matrix:



	C1	C2	C3	C4
R1	0	1	1	0
R2	1	0	1	1
R3	0	1	0	1
R4	0	0	1	0
R5	1	0	1	0
R6	0	1	0	0

Perform a minhashing of the data, with the order of rows: R4, R6, R1, R3, R5, R2.

Note: we give the minhash value in terms of the original name of the row, rather than the order of the row in the permutation. These two schemes are equivalent, since we only care whether hash values for two columns are equal, not what their actual values are.

Answer:

	C1	**C2**	**C3**	**C4**
R4			**R4**	
R6		**R6**	**R4**	
R1		**R6**	**R4**	
R3		**R6**	**R4**	**R3**
R5	**R5**	**R6**	**R4**	**R3**
R2	**R5**	**R6**	**R4**	**R3**

Distance Measures

Question 1:

Consider the following three vectors u, v, w in a 6-dimensional space:

u = [1, 0.25, 0, 0, 0.5, 0]
v = [0.75, 0, 0, 0.2, 0.4, 0]
w = [0, 0.1, 0.75, 0, 0, 1]

Suppose cos(x,y) denotes the similarity of vectors x and y under the cosine similarity measure. Compute all three pairwise similarities among u,v, w.

Answer:

Given data is:

u = [1, 0.25, 0, 0, 0.5, 0]
v = [0.75, 0, 0, 0.2, 0.4, 0]
w = [0, 0.1, 0.75, 0, 0, 1]

|u| = $\sqrt{\mathbf{1^2 + 0.25^2}} + \mathbf{0^2 + 0^2 + 0.5^2 + 0^2 = 1.145}$

|v| = $\sqrt{\mathbf{0.75^2 + 0^2}} + \mathbf{0^2 + 0.2^2 + 0.4^2 + 0^2 = 0.873}$

|w| = $\sqrt{\mathbf{0^2 + 0.1^2}} + \mathbf{0.75^2 + 0^2 + 0^2 + 1^2 = 1.25}$

cos (u, v) = $\frac{\mathrm{u}^*\mathrm{v}}{|u|*|v|} = \frac{0.75 + 0.02}{1.145 * 0.873} \rightarrow \boldsymbol{\theta} = 18\, degrees.$

cos (v, w) = $\frac{\mathrm{v}^*\mathrm{w}}{|\mathrm{v}|*|\mathrm{w}|} = \frac{0}{0.873^*1.25} \rightarrow \boldsymbol{\theta} = 0\, degrees.$

cos (u, w) = $\frac{\mathrm{u}^*\mathrm{w}}{|u|*|\mathrm{w}|} = \frac{0.025}{1.145 * 1.25} \rightarrow \boldsymbol{\theta} = 89\, degrees.$

Question 2:

Here are five vectors in a 10-dimensional space:
1111000000 0100100101 0000011110 0111111111 1011111111
Compute the Jaccard distance (not Jaccard "measure") between each pair of the vectors.
Answer:
Let A = 1111000000; B = 0100100101, C = 0000011110, D = 0111111111, E = 1011111111
Jaccard Distance (A, B) = 1 – (1/7) = 6/7 **Jaccard Distance (A, C) = 1 – (0/8) = 1**
Jaccard Distance (A, D) = 1 – (3/10) = 7/10 **Jaccard Distance (A, E) = 1 – (3/10) = 7/10**
Jaccard Distance (B, C) = 1 – (1/7) = 6/7 **Jaccard Distance (B, D) = 1 – (4/9) = 5/9**
Jaccard Distance (B, E) = 1 – (3/10) = 7/10 **Jaccard Distance (C, D) = 1 – (4/9) = 5/9**
Jaccard Distance (C, E) = 1 – (4/9) = 5/9 **Jaccard Distance (D, E) = 1 – (8/10) = 2/10**

Question 3:

Here are five vectors in a 10-dimensional space:
1111000000 0100100101 0000011110 0111111111 1011111111
Compute the Manhattan distance (L_1 norm) between each two of these vectors.

Answer:

Let A = 1111000000; B = 0100100101, C = 0000011110, D = 0111111111, E = 1011111111

Manhattan distance of A, B = 6

Manhattan distance of A, C = 8

Manhattan distance of A, D = 7

Manhattan distance of A, E = 7

Manhattan distance of B, C = 6

Manhattan distance of B, D = 5

Manhattan distance of B, E = 7

Manhattan distance of C, D = 5

Manhattan distance of C, E = 5

Manhattan distance of D, E = 2

Question 4: The edit distance is the minimum number of character insertions and character deletions required to turn one string into another. Compute the edit distance between each pair of the strings **he**, **she**, **his**, and **hers**.

Answer:

The edit distance between he and she = 1

The edit distance between he and his = 3

The edit distance between he and hers = 2

The edit distance between she and his = 4

The edit distance between she and hers = 3

The edit distance between his and hers = 3

Frequent item sets

Question 1:

Suppose we have transactions that satisfy the following assumptions:

- *s*, the support threshold, is 10,000.
- There are one million items, which are represented by the integers 0,1,...,999999.
- There are *N* frequent items, that is, items that occur 10,000 times or more.
- There are one million pairs that occur 10,000 times or more.
- There are 2*M* pairs that occur exactly once. *M* of these pairs consist of two frequent items, the other *M* each have at least one nonfrequent item.
- No other pairs occur at all.
- Integers are always represented by 4 bytes.

Suppose we run the a-priori algorithm to find frequent pairs and can choose on the second pass between the triangular-matrix method for counting candidate pairs (a triangular array count[i][j] that holds an integer count for each pair of items (*i, j*) where *i* < *j*) and a hash table of item-item-count triples. Neglect in the first case the space needed to translate between original item numbers and numbers for the frequent items, and in the second case neglect the space needed for the hash table. Assume that item numbers and counts are always 4-byte integers.

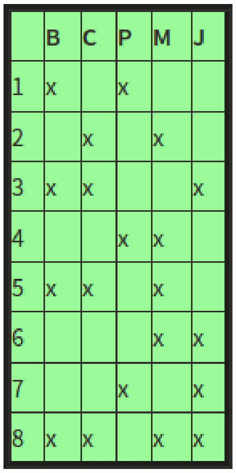
As a function of *N* and *M*, what is the minimum number of bytes of main memory needed to execute the a-priori algorithm on this data?

Answer:

One data structure is needed to hold the counts of each item. This will be an array of length 1,000,000 (A million items), which at 4 bytes an integer, is 4 million bytes. Keeping an array of length N will take up 4N bytes to keep the counts of the frequent items
A hash table is needed to hold M values. The two items in the pair and the count will be recorded so that 3 integers x 4 bytes = 12 bytes per integer, so the size of this will be 12M.
The minimum number of bytes of main memory needed to execute the a-priori algorithm on this data is S = 4N+12M

Question 2:

Below is a table representing eight transactions and five items: Beer, Coke, Pepsi, Milk, and Juice. The items are represented by their first letters; e.g., "M" = milk. An "x" indicates membership of the item in the transaction.



	B	C	P	M	J
1	x		x		
2		x		x	
3	x	x			x
4			x	x	
5	x	x		x	
6				x	x
7			x		x
8	x	x		x	x

Compute the support for each of the 10 pairs of items. If the support threshold is 2, find out the pairs that are frequent item sets.

Answer:

The pairs of item sets are:
(B, C), (C, M) → 3
(B, J), (B, M), (C, J), (M, J) → 2
(B, P), (P, J), (P, M) → 1
(C, P) → 0

Question 3:

Suppose we perform the PCY algorithm to find frequent pairs, with market-basket data meeting the following specifications:

- *s*, the support threshold, is 10,000.
- There are one million items, which are represented by the integers 0,1,...,999999.
- There are 250,000 frequent items, that is, items that occur 10,000 times or more.
- There are one million pairs that occur 10,000 times or more.
- There are *P* pairs that occur exactly once and consist of 2 frequent items.
- No other pairs occur at all.
- Integers are always represented by 4 bytes.
- When we hash pairs, they distribute among buckets randomly, but as evenly as possible; i.e., you may assume that each bucket gets exactly its fair share of the *P* pairs that occur once.

Suppose there are *S* bytes of main memory. In order to run the PCY algorithm successfully, the number of buckets must be sufficiently large that most buckets are not frequent. In addition, on the second pass, there must be enough room to count all the candidate pairs. As a function of *S*, what is the largest value of *P* for which we can successfully run the PCY algorithm on this data? Find out the value for *S* and value for *P* that is approximately (i.e., to within 10%) the largest possible value of *P* for that *S*.

Answer:

S = 10,000; Items = 1,000,000.
P = 1000000 / buckets
Number of frequent pairs that map to a bucket = p * (1000000/buckets)
During pass 1, we have at most (S- 4MB) / 4 ~~ S/4 buckets
For second pass, we need P * 12000000 / buckets
In order for there to be enough space for all these counts, we need S >= 48,000,000P/S, or P <= S^2 /48,000,000.

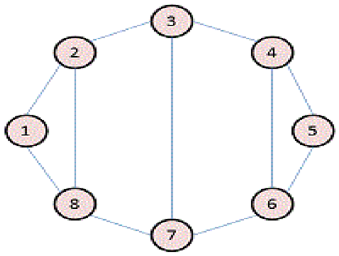
Question 4: During a run of Toivonen's Algorithm with set of items {A,B,C,D,E,F,G,H} a sample is found to have the following maximal frequent itemsets: {A,B}, {A,C}, {A,D}, {B,C}, {E}, {F}. Compute the negative border.

Answer:
The negative border consists of fourteen sets: {G}, {H}, {A, E}, {A, F}, {B, D}, {B, E}, {B, F}, {G, D}, {C, E}, {C, F}, {D, E}, {D, F}, {E, F}, {A, B, C}.

Communities

Question 1:

For the following graph:



Write the adjacency matrix A, the degree matrix D, and the Laplacian matrix L. For each, find the sum of all entries and the number of nonzero entries.

Answer:

Adjacency Matrix (A):

	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	1
2	1	0	1	0	0	0	0	1
3	0	1	0	1	0	0	1	0
4	0	0	1	0	1	1	0	0
5	0	0	0	1	1	0	0	0
6	0	0	0	1	1	0	1	0
7	0	0	1	0	0	1	0	1
8	1	1	0	0	0	0	1	0

No of non-zero entries = 22; Sum of all entries = 22

Degree Matrix (D):

	1	2	3	4	5	6	7	8
1	2	0	0	0	0	0	0	0
2	0	3	0	0	0	0	0	0
3	0	0	3	0	0	0	0	0
4	0	0	0	3	0	0	0	0
5	0	0	0	0	2	0	0	0
6	0	0	0	0	0	3	0	0
7	0	0	0	0	0	0	3	0
8	0	0	0	0	0	0	0	3

No of non-zero entries = 8; Sum of all entries = 8

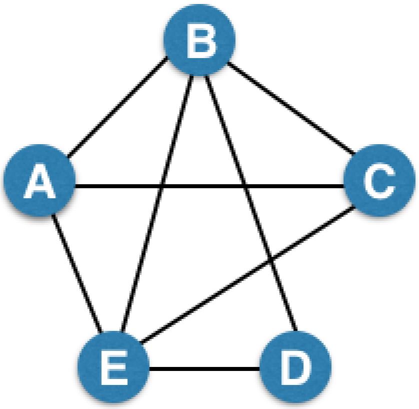
Laplacian Matrix is L = D – A

	1	2	3	4	5	6	7	8
1	2	-1	0	0	0	0	0	-1
2	-1	3	-1	0	0	0	0	-1
3	0	-1	3	-1	0	0	-1	0
4	0	0	-1	3	-1	-1	0	0
5	0	0	0	-1	2	-1	0	0
6	0	0	0	-1	-1	3	-1	0
7	0	0	-1	0	0	-1	3	-1
8	-1	-1	0	0	0	0	-1	3

No of non-zero entries = 30; Sum of all entries = 0

Question 2:

Consider the following undirected graph (i.e., edges may be considered bidirectional):

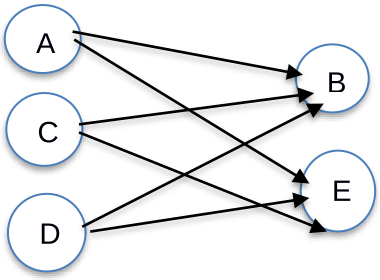


Run the "trawling" algorithm for finding dense communities on this graph and find all complete bipartite subgraphs of types $K_{3,2}$ and $K_{2,2}$. Note: In the case of $K_{2,2}$, we consider {{W, X}, {Y, Z}} and {{Y, Z}, {W, X}} to be identical.

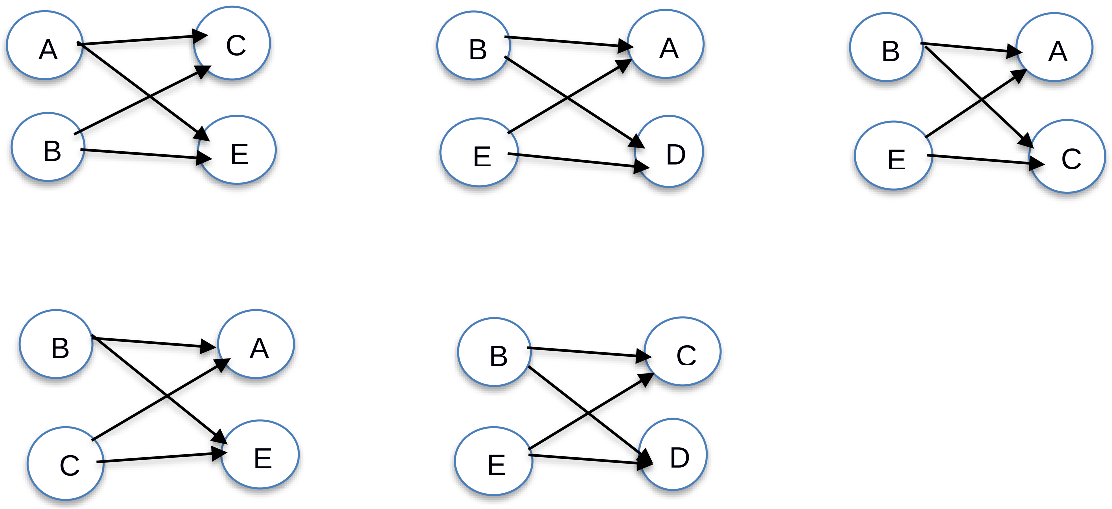
Answer:

From the given graph, A = {B, C, E}; B = {A, C, D, E}; C = {A, B, E}; D = {B, E}; E = {A, B, C, D}. Here B and E are having support more than A, C, D.

Bipartite subgraph of $K_{3,2}$ is:

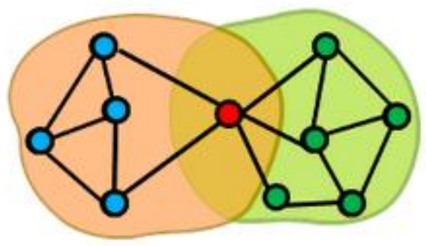


Bipartite subgraph of $K_{2,2}$ is:

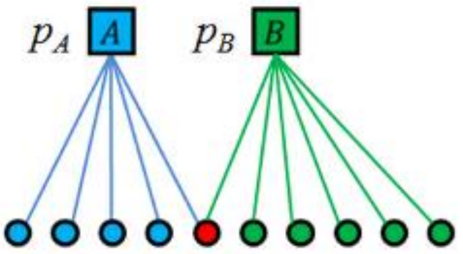


Question 3:

We fit AGM to the network on the left, and found the parameters on the right:



Network



Learned AGM parameters

Find the optimal values for p_A and p_B.

Answer:

Pa = Number of edges in the network / Total possible number of edges = 7/5c2 = 7/10.

Pb = Number of edges in the network / Total possible number of edges = 9/6c2 = 9/15.

Stream Algorithms

Question 1: We wish to estimate the surprise number (2nd moment) of a data stream, using the method of AMS. It happens that our stream consists of ten different values, which we'll call 1, 2,..., 10, that cycle repeatedly. That is, at timestamps 1 through 10, the element of the stream equals the timestamp, at timestamps 11 through 20, the element is the timestamp minus 10, and so on. It is now timestamp 75, and a 5 has just been read from the stream. As a start, you should calculate the surprise number for this time.

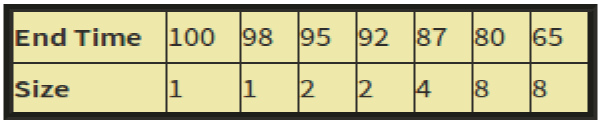
For our estimate of the surprise number, we shall choose three timestamps at random, and estimate the surprise number from each, using the AMS approach (length of the stream times 2*m*-1, where *m* is the number of occurrences of the element of the stream at that timestamp, considering all times from that timestamp on, to the current time). Then, our estimate will be the median of the three resulting values.

You should discover the simple rules that determine the estimate derived from any given timestamp and from any set of three timestamps. Then, take any 4 examples of the set of three "random" timestamps, find out the closest estimate among the 4 examples.

Answer:

First, the surprise number is 5*64 + 5*49 = 565. The reason is that the elements 1 to 5 appear 8 times, so they contribute $5*8^2$, and the elements 6 to 10 appear 7 times, contributing $5*7^2$. The AMS estimate is a nondecreasing function of the timestamp. Thus, of any three timestamps, the middle one will give the median estimate, and we do not have to calculate all three. At each of the timestamps between 36 and 45, inclusive, the element appearing then appears exactly 4 times, from that time forward. Thus, each of these timestamps generates an estimate of 75*(2*4 - 1) = 525, which is as close to 565 as we can get. Each of the correct answers has a middle timestamp in this range. Similarly, for the timestamps between 26 and 35, the estimate is 75*(2*5 - 1) = 675 and for the timestamps between 46 and 55 the estimate is 75*(2*3 - 1) = 375. Neither of these groups offer as close an estimate, and the timestamps earlier or later offer even worse estimates.

Question 2: Suppose we are using the DGIM algorithm of Section 4.6.2 to estimate the number of 1's in suffixes of a sliding window of length 40. The current timestamp is 100, and we have the following buckets stored:



End Time	100	98	95	92	87	80	65
Size	1	1	2	2	4	8	8

Note: we are showing timestamps as absolute values, rather than modulo the window size, as DGIM would do. Suppose that at times 101 through 105, 1's appear in the stream. Compute the set of buckets that would exist in the system at time 105. Buckets are represented by pairs (end-time, size).

Answer: Process: -

1. **Add +1 on timeline, check if there are more than 2 buckets with same size.**
2. **If there are more than 2 buckets with same size, group them to a new size=sizex2 and with timestamp equals to the latest one.**
3. **Keep checking until there are no more than 2 buckets with same size.**

By doing the above process on the given data, we get finally the set of buckets that would exist in the system at time 105 as

{(105,1), (104,2), (102,4), (95, 8), (80, 16)}

Question 3: We wish to use the Flagolet-Martin algorithm of Section 4.4 to count the number of distinct elements in a stream. Suppose that there are ten possible elements, 1, 2,..., 10, that could appear in the stream, but only four of them have actually appeared. To make our estimate of the count of distinct elements, we hash each element to a 4-bit binary number. The element *x* is hashed to 3*x* + 7 (modulo 11). For example, element 8 hashes to 3*8+7 = 31, which is 9 modulo 11 (i.e., the remainder of 31/11 is 9). Thus, the 4-bit string for element 8 is 1001.

A set of four of the elements 1 through 10 could give an estimate that is exact (if the estimate is 4), or too high, or too low. You should figure out under what circumstances a set of four elements falls into each of those categories. Then, take any 4 examples of the set of four elements, find out the exactly correct estimate among 4 examples.

Answer: Given hash function h(x) = 3*x* + 7 (modulo 11).

x	h(x)	Binary format of h(x)	Trailing Zeroes
1	10	1010	1
2	2	0010	1
3	5	0101	0
4	8	1000	3
5	0	0000	4
6	3	0011	0
7	6	0110	1
8	9	1001	0
9	1	0001	0
10	4	0100	2

Question 4: A certain Web mail service (like gmail, e.g.) has 10^8 users, and wishes to create a sample of data about these users, occupying 10^{10} bytes. Activity at the service can be viewed as a stream of elements, each of which is an email. The element contains the ID of the sender, which must be one of the 10^8 users of the service, and other information, e.g., the recipient(s), and contents of the message. The plan is to pick a subset of the users and collect in the 10^{10} bytes records of length 100 bytes about every email sent by the users in the selected set (and nothing about other users).

The method of Section 4.2.4 will be used. User ID's will be hashed to a bucket number, from 0 to 999,999. At all times, there will be a threshold t such that the 100-byte records for all the users whose ID's hash to t or less will be retained, and other users' records will not be retained. You may assume that each user generates emails at exactly the same rate as other users. As a function of n, the number of emails in the stream so far, what should the threshold t be in order that the selected records will not exceed the 10^{10} bytes available to store records?

Answer:

Suppose that the fraction of users in the sample is p.
The number of users whose records are stored is 10^8.
Since each user generates 10^8 emails in the stream when n emails have been seen, then the number of records stored is $10^8 * p * 10^{-8}$ = pn.
Since each record is 100 bytes, we can store $10^{10}/100 = 10^8$ records.
That is, pn = 10^8, or p = $10^8/n$.
If the threshold is t, the fraction p of users that will be in the selected set is (t+1)/1,000,000 = $10^8/n$.
Therefore, t = $(10^{14} / n) - 1$

Question 5: Suppose we hash the elements of a set S having 23 members, to a bit array of length 100. The array is initially all-0's, and we set a bit to 1 whenever a member of S hashes to it. The hash function is random and uniform in its distribution. What is the expected fraction of 0's in the array after hashing? What is the expected fraction of 1's? You may assume that 100 is large enough that asymptotic limits are reached.

Answer:

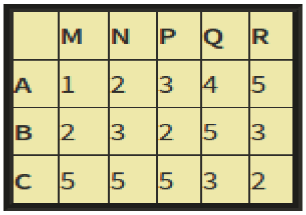
Members: 23; Bit array of length: 100; T = 100; hash function = 1

Expected fraction of 0's = $e^{-hd}/t = e^{-23}/100$

Expected fraction of 1's = 1 - e^{-hd}/t = 1 - $e^{-23}/100$

Recommendation Systems

Question 1: Here is a table of 1-5-star ratings for five movies (M, N, P. Q. R) by three raters (A, B, C).



	M	N	P	Q	R
A	1	2	3	4	5
B	2	3	2	5	3
C	5	5	5	3	2

Normalize the ratings by subtracting the average for each row and then subtracting the average for each column in the resulting table. Then, identify largest element and entry of (C, P) about the normalized table.

Answer:

Average of A = 15/5 = 3; Average of B = 15/5 = 3; Average of C = 20/5 = 4
Subtracting the average for each row, we get

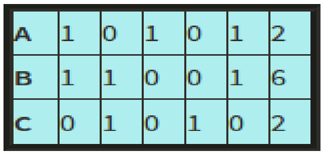
	M	N	P	Q	R
A	**-2**	**-1**	**0**	**1**	**2**
B	**-1**	**0**	**-1**	**2**	**0**
C	**1**	**1**	**1**	**-1**	**-2**

Average of M = -2/3 = -0.66; Average of N = 0; Average of P = 0; Average of Q = 2/3 = 0.66; Average of R = 0
Subtracting the average for each column, we get

	M	N	P	Q	R
A	**-4/3**	**-1**	**0**	**1/3**	**2**
B	**-1/3**	**0**	**-1**	**4/3**	**0**
C	**5/3**	**1**	**1**	**-5/3**	**-2**

Largest Element is 2 and Entry of (C, P) is 1

Question 2: Below is a table giving the profile of three items.



A	1	0	1	0	1	2
B	1	1	0	0	1	6
C	0	1	0	1	0	2

The first five attributes are
Boolean, and the last is an integer "rating." Assume that the scale factor for the rating is α. Compute, as a function of α, the cosine distances between each pair of profiles. For each of α = 0, 0.5, 1, and 2, determine the cosine of the angle between each pair of vectors.

Answer:

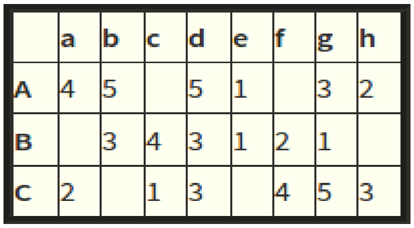
cos (A, B) = $\frac{A*B}{|A|*|B|} = \frac{1+1+12\propto^2}{\sqrt{3+4\propto^2}\sqrt{3+36\propto^2}} = \frac{2+12\propto^2}{\sqrt{9+120\propto^2+144\propto^4}}$

cos (B, C) = $\frac{B*C}{|B|*|C|} = \frac{1+12\propto^2}{\sqrt{3+36\propto^2}\sqrt{2+4\propto^2}} = \frac{1+12\propto^2}{\sqrt{6+84\propto^2+144\propto^4}}$

cos (C, A) = $\frac{C*A}{|C|*|A|} = \frac{4\propto^2}{\sqrt{3+4\propto^2}\sqrt{2+4\propto^2}} = \frac{4\propto^2}{\sqrt{6+20\propto^2+16\propto^4}}$

If α = 0 → cos (A, B) = 0.66, cos (B, C) = 0.408, cos (C, A) = 0
If α = 0.5 → cos (A, B) = 0.7216, cos (B, C) = 0.6667, cos (C, A) = 0.28868
If α = 1 → cos (A, B) = 0.8473, cos (B, C) = 0.8498, cos (C, A) = 0.6172
If α = 2 → cos (A, B) = 0.9461, cos (B, C) = 0.9926, cos (C, A) = 0.8652

Question 3: Below is a utility matrix representing ratings by users A, B, and C for items *a* through h.



	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Treat ratings of 3, 4, and 5 as 1 and 1, 2, and blank as 0. Compute the Jaccard distance between each pair of items. Then, cluster the items hierarchically into four clusters, using the Jaccard distance. When a cluster consists of more than one item, take the distance between clusters to be the minimum over all pairs of items, one from each cluster, of the Jaccard distance between those items. Break ties lexicographically. That is, sort the items that would be merged alphabetically, and merge those clusters whose resulting set would be first alphabetically.

Note: if you are not familiar with hierarchical clustering, read Sect. 7.2 of the MMDS book.

Answer:

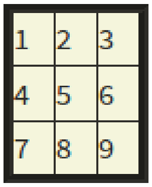
Update matrix by replacing 3, 4 and 5 as 1 and 1, 2 and blank as 0

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**
A	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**
B	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
C	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**

Jaccard Distance of a, b is 1- ½ = ½
Jaccard Distance of a, c is 1- 0 = 1
Jaccard Distance of a, d is 1- 1/3 = 1/3
Jaccard Distance of a, e is 1- 0 = 1
Jaccard Distance of a, f is 1- 0 = 0
Jaccard Distance of a, g is 1- ½ = ½

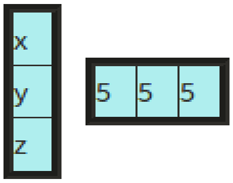
Jaccard Distance of a, h is 1- 0 = 0
Jaccard Distance of b, c is 1- ½ = ½
Jaccard Distance of b, d is 1- 2/3 = 1/3
Jaccard Distance of b, e is 1- 0 = 1
Jaccard Distance of b, f is 1- 0 = 1
Jaccard Distance of b, g is 1- 1/3 = 2/3
Jaccard Distance of b, h is 1- 0 = 1
Jaccard Distance of c, d is 1- 1/3 = 2/3
Jaccard Distance of c, e is 1- 0 = 1
Jaccard Distance of c, f is 1- 0 = 1
Jaccard Distance of c, g is 1- 0 = 1
Jaccard Distance of c, h is 1- 0 = 1
Jaccard Distance of d, e is 1- 0 = 1
Jaccard Distance of d, f is 1- 1/3 = 2/3
Jaccard Distance of d, g is 1- 2/3 = 1/3
Jaccard Distance of d, h is 1- 1/3 = 2/3
Jaccard Distance of e, f is 1- 0 = 1
Jaccard Distance of e, g is 1- 0 = 1
Jaccard Distance of e, h is 1- 0 = 1
Jaccard Distance of f, g is 1- ½ = ½
Jaccard Distance of f, h is 1- 0 = 1
Jaccard Distance of g, h is 1- ½ = ½
The final clusters are {f, h}, {b, d, g, a}, {c}, {e}

Question 4: We want to do an approximate UV-decomposition of the matrix M =



1	2	3
4	5	6
7	8	9

We shall use only a single column for U and a single row for V, so the goal is to make the product UV as close as possible to M. Initially, we shall set V to [5,5,5] and make the entries of U unknown. Then in the first step, we choose the values of x, y, and z that minimize the root-mean-square error (RMSE) between the product



x
y
z

5	5	5

and the matrix M. Find the values of x, y, and z that minimize the RMSE

Answer:
Given matrices M, U, V

The product of U and V is $\begin{matrix} 5x & 5x & 5x \\ 5y & 5y & 5y \\ 5z & 5z & 5z \end{matrix}$

Let the equation be $(5x-1)^2 + (5x-2)^2 + (5x-3)^2 + (5y-4)^2 + (5y-5)^2 + (5y-6)^2 + (5z-7)^2 + (5z-8)^2 + (5z-9)^2 = 0$
Differentiate above equation with x, we get 150x – 60 = 0 → x = 2/5
Differentiate above equation with y, we get 150y – 150 = 0 → y = 1
Differentiate above equation with z, we get 150z – 240 = 0 → z = 8/5

Dimensionality Reduction

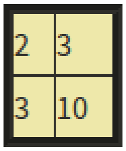
Question 1: Note: In this question, all columns will be written in their transposed form, as rows, to make the typography simpler. Matrix M has three rows and three columns, and the columns form an orthonormal basis. One of the columns is [2/7,3/7,6/7], and another is [6/7, 2/7, -3/7]. Let the third column be [x,y,z]. Since the length of the vector [x,y,z] must be 1, there is a constraint that $x^2+y^2+z^2 = 1$. However, there are other constraints, and these other constraints can be used to deduce facts about the ratios among x, y, and z. Compute these ratios.

Answer:

Let C1 be [2/7,3/7,6/7], C2 be [6/7, 2/7, -3/7] and C3 be [x, y, z]
The dot product of any two columns must be zero.
C1.C2 = (2/7 * 6/7) + (3/7 * 2/7) + (6/7 * -3/7) = 0
C2.C3 = (6/7 * x) + (2/7 * y) + (-3/7 * z) = 0 → 6x +2y -3z = 0 – Eq 1
C3.C1 = (x * 2/7) + (y * 3/7) + (z * 6/7) = 0 → 2x + 3y + 6z = 0 – Eq 2
2 * Eq 1 + Eq 2 → 12x + 4y -6z + 2x + 3y +6z = 0 → 14x + 7y = 0 → y = -2x
3 * Eq 2 – Eq 1 → 6x + 9y + 18z – 6x – 2y + 3z = 0 → 7y + 21z = 0 → y = -3z
x: y: z = -2: 1: -3

Question 2: Find the eigenvalues and eigenvectors of the following matrix:

You should assume the first component of an eigenvector is 1. Then, find out One eigenvalue and One eigenvector.



2	3
3	10

Answer:

Let the given matrix be A = $\begin{matrix} 2 & 3 \\ 3 & 10 \end{matrix}$ **and the eigen vector be of the form** $\begin{matrix} 1 \\ e \end{matrix}$

Ax = λx ➔ $\begin{matrix} 2 & 3 \\ 3 & 10 \end{matrix} * \begin{matrix} 1 \\ e \end{matrix}$ **= λ *** $\begin{matrix} 1 \\ e \end{matrix}$ **→ 2 + 3e = λ and 3 + 10e = λe → 3 + 10e = (2 + 3e)e**
$3e^2$ – 8e + 3 = 0 → e = 3, -1/3

The eigen vectors are $\begin{matrix} \mathbf{1} \\ \mathbf{3} \end{matrix} \boldsymbol{and} \begin{matrix} \mathbf{1} \\ \mathbf{-1/3} \end{matrix}$
The eigen values are 2 + 3e = λ → λ = 2 + 3*3 = 11 and λ = 2 + 3*(-1/3) = 1

Question 3: Suppose [1,3,4,5,7] is an eigenvector of some matrix. What is the unit eigenvector in the same direction? Find out the components of the unit eigenvector.

Answer:

Given the eigen vector of some matrix be M = [1,3,4,5,7]
To get the unit eigen vector of given matrix, we need to divide each component by square root of sum of squares in the same direction.
Sum of squares = $1^2 + 3^2 + 4^2 + 5^2 + 7^2$ = 100 and its square root is 10
Unit Eigen Vector = [1/10,3/10,4/10,5/10,7/10]

Question 4: Suppose we have three points in a two-dimensional space: (1,1), (2,2), and (3,4). We want to perform PCA on these points, so we construct a 2-by-2 matrix, call it N, whose eigenvectors are the directions that best represent these three points. Construct the matrix N and identify, its elements.

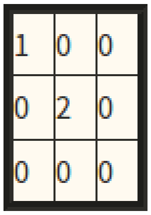
Answer:

The given three points in a 2- D space are (1,1), (2,2), and (3,4).

We should construct a matrix whose rows correspond to points and columns correspond to dimensions of the space.

Then the matrix will be M = $\begin{matrix} 1 & 1 \\ 2 & 2 \\ 3 & 4 \end{matrix}$ **M^T M =** $\begin{matrix} 1 & 2 & 3 \\ 1 & 2 & 4 \end{matrix}$ ***** $\begin{matrix} 1 & 1 \\ 2 & 2 \\ 3 & 4 \end{matrix}$ **=** $\begin{matrix} 14 & 17 \\ 17 & 21 \end{matrix}$

Question 5: Consider the diagonal matrix M =



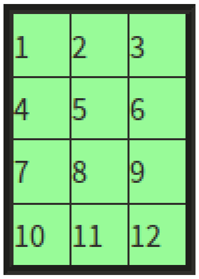
1	0	0
0	2	0
0	0	0

Compute its Moore-Penrose pseudoinverse.

Answer:

Moore-Penrose pseudoinverse means the matrix having diagonal elements replaced by 1 and divided by corresponding elements of given matrix and the other elements will be zero. Moore-Penrose pseudoinverse of given matrix is $\begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix}$

Question 6: When we perform a CUR decomposition of a matrix, we select rows and columns by using a particular probability distribution for the rows and another for the columns. Here is a matrix that we wish to decompose:



1	2	3
4	5	6
7	8	9
10	11	12

Calculate the probability distribution for the rows.

Answer:

Probability with which we choose now = $\frac{\text{sum of squares of elements in the rows}}{\text{sum of squares of elements in the matrix}}$

Sum of squares of elements in the matrix = 12*13*25/6 = 3900/6 = 650

P(R1) = $\frac{1^2 + 2^2 + 3^2}{650}$ **= 14/650 = 0.02**

P(R2) = $\frac{4^2 + 5^2 + 6^2}{650}$ **= 77/650 = 0.12**

P(R3) = $\frac{7^2 + 8^2 + 9^2}{650}$ **= 194/650 = 0.298**

P(R4) = $\frac{10^2 + 11^2 + 12^2}{650}$ **= 365/650 =0.56**

Clustering

Question 1: We can cluster in one dimension as well as in many dimensions. In this problem, we are going to cluster numbers on the real line. The particular numbers (data points) are 1, 4, 9, 16, 25, 36, 49, 64, 81, and 100, i.e., the squares of 1 through 10. We shall use a k-means algorithm, with two clusters. You can verify easily that no matter which two points we choose as the initial centroids, some prefix of the sequence of squares will go into the cluster of the smaller and the remaining suffix goes into the other cluster. As a result, there are only nine different clusterings that can be achieved, ranging from {1} {4, 9, ...,100} through {1, 4, ...,81}{100}.

We then go through a reclustering phase, where the centroids of the two clusters are recalculated and all points are reassigned to the nearer of the two new centroids. For each of the nine possible clusterings, calculate how many points are reclassified during the reclustering phase. List out pair of initial centroids that results in *exactly one* point being reclassified.

Answer:

Given data points are 1, 4, 9, 16, 25, 36, 49, 64, 81, and 100.
There can be 9 different clusters as follows:

1. **{1}, (4, 9, 16, 25, 36, 49, 64, 81, 100},**
2. **{1, 4}, {9, 16, 25, 36, 49, 64, 81, 100},**
3. **{1, 4, 9}, {16, 25, 36, 49, 64, 81, 100},**
4. **{1, 4, 9, 16}, {25, 36, 49, 64, 81, 100},**
5. **{1, 4, 9, 16, 25}, {36, 49, 64, 81, 100},**
6. **{1, 4, 9, 16, 25, 36}, {49, 64, 81, 100},**
7. **{1, 4, 9, 16, 25, 36, 49}, {64, 81, 100},**
8. **{1, 4, 9, 16, 25, 36, 49, 64}, {81, 100},**
9. **{1, 4, 9, 16, 25, 36, 49, 64, 81}, {100}.**

Only one point has to be shifted between clusters if we change the centroid. Let the initial values be 36 and 100. Mean of 36 and 100 = (36+100)/2 = 68.
So, the clusters will be {1, 4, 9, 16, 25, 36, 49, 64}, {81, 100}.
Centroids of these clusters are 25.5 and 90.5. Mean = (25.5 + 90.5)/2 = 58.
Now the clusters will be {1, 4, 9, 16, 25, 36, 49}, {64, 81, 100}. Here the only one element is shifted between clusters.

Question 2: Suppose we want to assign points to one of two cluster centroids, either (0,0) or (100,40). Depending on whether we use the L_1 or L_2 norm, a point (x,y) could be clustered with a different one of these two centroids. For this problem, you should work out the conditions under which a point will be clustered with the centroid (0,0) when the L_1 norm is used, but clustered with the centroid (100,40) when the L_2 norm is used. List out those points.

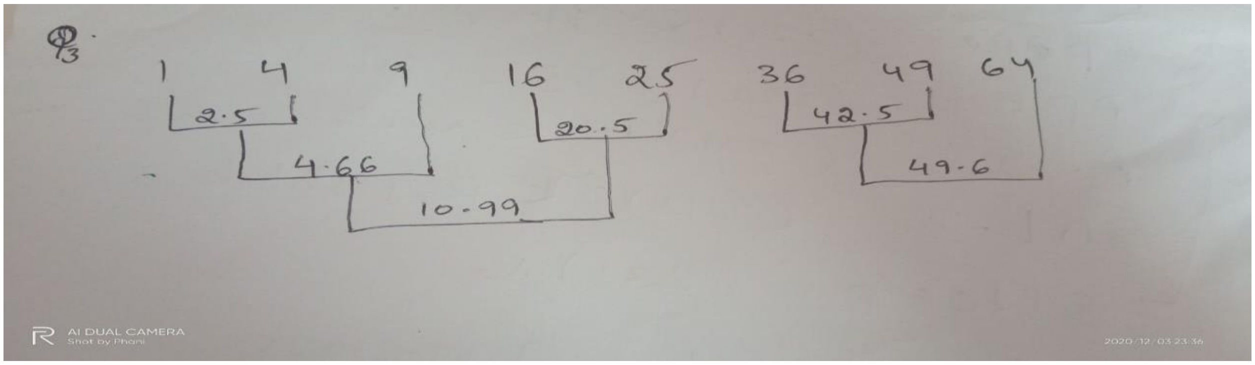
Answer:

Given centroids are (0,0), (100, 40).

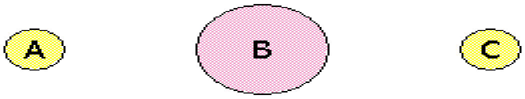
Given a point (x, y) which could be clustered with a different one of these two centroids.
L1 norm is the Manhattan Distance and L2 norm is the Euclidean Distance.
After L1 norm and L2 norm are calculated the values of x and y are 55, 5.
When L1 norm is applied on point (55, 5), the point is clustered with centroid (0, 0).
When L2 norm is applied on point (55, 5), the point is clustered with centroid (100, 40).

Question 3: Suppose our data set consists of the perfect squares 1, 4, 9, 16, 25, 36, 49, and 64, which are points in one dimension. Perform a hierarchical clustering on these points, as follows. Initially, each point is in a cluster by itself. At each step, merge the two clusters with the closest centroids, and continue until only two clusters remain. Which centroid of a cluster that exists at some time during this process? Positions are represented to the nearest 0.1.

Answer:



Question 4: Suppose that the true data consists of three clusters, as suggested by the diagram below:



There is a large cluster B centered around the origin (0,0), with 8000 points uniformly distributed in a circle of radius 2. There are two small clusters, A and C, each with 1000 points uniformly distributed in a circle of radius 1. The center of A is at (-10,0) and the center of C is at (10,0).

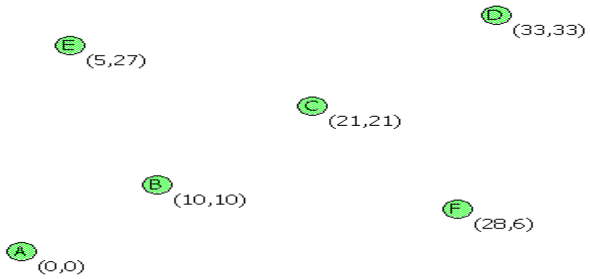
Suppose we choose three initial centroids x, y, and z, and cluster the points according to which of x, y, or z they are closest. The result will be three *apparent* clusters, which may or may not coincide with the *true* clusters A, B, and C. Say that one of the true clusters is *correct* if there is an apparent cluster that consists of all and only the points in that true cluster. Assuming initial centroids x, y, and z are chosen independently and at random, what is the probability that A is correct? What is the probability that C is correct? What is the probability that both are correct?

Answer:

Given centroids are x, y, z

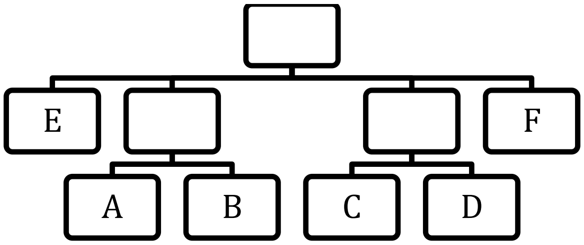
We can assign each of x, y, z to A, B, C in 27 possible ways.
Chance of being in A is 1000/10000 = 0.1
Chance of being in B is 8000/10000 = 0.8
Chance of being in C is 1000/10000 = 0.1
There are 6 different cases to interchange x, y, z in A, B, C which will be total 27.
The probability that A is correct is 24%
The probability that C is correct is 24%
The probability that A & C are correct is 4.8%

Question 5: Perform a hierarchical clustering of the following six points:



using the *complete-link* proximity measure (the distance between two clusters is the largest distance between any two points, one from each cluster). Find out a cluster at some stage of the agglomeration?

Answer:



	A	**B**	**C**	**D**	**E**	**F**
A	**0**	**14.1**	**29.6**	**46.6**	**27.4**	**28.6**
B		**0**	**15.5**	**39.5**	**17.7**	**18.4**
C			**0**	**16.9**	**17**	**16.5**
D				**0**	**28.5**	**27.4**
E					**0**	**31.1**
F						**0**

As A and B are low, clustering will be done as follows:
A and B will be clustered with C → AC – 29.6 and BC – 15.5
A and B will be clustered with D → AD – 46.6 and BD – 32.5
A and B will be clustered with E → AE – 27.4 and BE – 17.7
A and B will be clustered with F → AF – 28.6 and BF – 18.4

C and D will be clustered → CD – 16.9
D and E will be clustered → DE – 28.6
D and F will be clustered → DF – 27.4
In CD, DE, DF as CD is low, clustering as follows:
C and D will be clustered with E → CE – 17 and DE – 28.6
C and D will be clustered with F → CF – 16.5 and DE – 27.4

Computational Advertising

Question 1: Suppose we apply the BALANCE algorithm with bids of 0 or 1 only, to a situation where advertiser A bids on query words x and y, while advertiser B bids on query words x and z. Both have a budget of $2. Identify a sequence of four queries that will certainly be handled optimally by the algorithm.

Answer:

yzyy is one sequence of four queries which yields the optimum i.e., $3.

yyxx is another sequence of four queries in which one advertiser is assigned to first x and the other is assigned to second x.

Question 2: The *set cover* problem is: given a list of sets, find a smallest collection of these sets such that every element in any of the sets is in at least one set of the collection. As we form a collection, we say an element is *covered* if it is in at least one set of the collection.

Note: In this problem, we shall represent sets by concatenating their elements, without brackets or commas. For example, {A, B} will be represented simply as AB.

There are many greedy algorithms that could be used to pick a collection of sets that is close to as small as possible. Here are some that you will consider in this problem.

Dumb: Select sets for the collection in the order in which they appear on the list. Stop when all elements are covered.

Simple: Consider sets in the order in which they appear on the list. When it is considered, select a set if it has at least one element that is not already covered. Stop when all elements are covered.

Largest-First: Consider sets in order of their size. If there are ties, break the tie in favor of the one that appears first on the list. When it is considered, select a set if it has at least one element that is not already covered. Stop when all elements are covered.

Most-Help: Consider sets in order of the number of elements they contain that are not already covered. If there are ties, break the tie in favor of the one that appears first on the list. Stop when all elements are covered.

Here is a list of sets:

AB, BC, CD, DE, EF, FG, GH, AH, ADG, ADF

First, determine the optimum solution, that is, the fewest sets that can be selected for a collection that covers all eight elements A, B, ..., H. Then, determine the sizes of the collections that will be constructed by each of the four algorithms mentioned above. Compute the ratio of the size returned by each algorithm to the optimum size.

Answer:

Given sets are: AB, BC, CD, DE, EF, FG, GH, AH, ADG, ADF

Dumb Method: AB, BC, CD, DE, EF, FG, GH → 7

Simple Method: AB, BC, CD, DE, EF, FG, GH → 7

Largest-First Method: ADG, ADF, AB, BC, DE, GH → 6

Most-Help Method: ADG, BC, EF, AH → 4

The optimum one is Most Help Method whose size is 4.

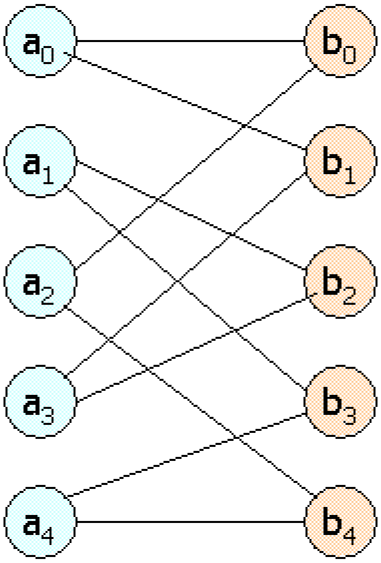
Ratio for Dumb and Most Help Methods = 7/4 = 1.75

Ratio for Simple and Most Help Methods = 7/4 = 1.75

Ratio for Largest First and Most Help Methods = 7/4 = 1.75

Ratio for Most Help and Most Help Methods = 7/4 = 1.75

Question 3: This bipartite graph:



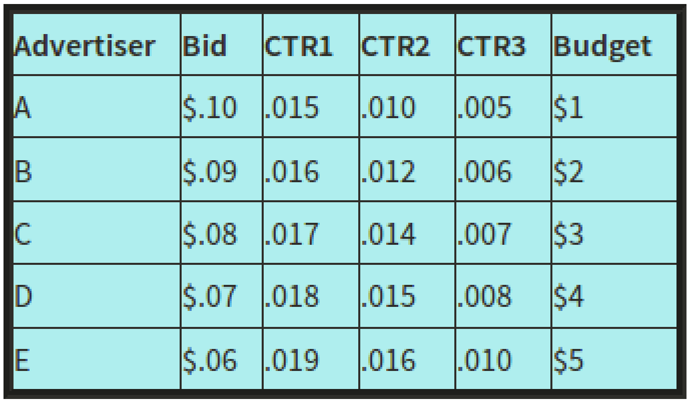
Has several perfect matchings. Find all the perfect matchings.

Answer:

The perfect matchings are: a0-b0, a1-b2, a4-b3, a3-b1, a2-b4, a0-b1, a1-b3, a2-b0, a3-b2, a4-b4.

Question 4: An ad publisher selects three ads to place on each page, in order from the top. Click-through rates (CTR's) at each position differ for each advertiser, and each advertiser has a different CTR for each position. Each advertiser bids for click-throughs, and each advertiser has a daily budget, which may not be exceeded. When a click-through occurs, the advertiser pays the amount they bid. In one day, there are 101 click-throughs to be auctioned.

Here is a table of the bids, CTR's for positions 1, 2, and 3, and budget for each advertiser.



Advertiser	Bid	CTR1	CTR2	CTR3	Budget
A	$.10	.015	.010	.005	$1
B	$.09	.016	.012	.006	$2
C	$.08	.017	.014	.007	$3
D	$.07	.018	.015	.008	$4
E	$.06	.019	.016	.010	$5

The publisher uses the following strategy to allocate the three ad slots:

1. Any advertiser whose budget is spent is ignored in what follows.
2. The first slot goes to the advertiser whose expected yield for the first slot (product of the bid and the CTR for the first slot) is the greatest. This advertiser is ignored in what follows.
3. The second slot goes to the advertiser whose expected yield for the second slot (product of the bid and the CTR for the second slot) is the greatest. This advertiser is ignored in what follows.
4. The third slot goes to the advertiser whose expected yield for the third slot (product of the bid and the CTR for the third slot) is the greatest.

The same three advertisers get the three ad positions until one of two things happens:

1. An advertiser runs out of budget, or
2. All 101 click-throughs have been obtained.

Either of these events ends one *phase* of the allocation. If a phase ends because an advertiser ran out of budget, then they are assumed to get all the clicks their budget buys. During the same phase, we calculate the number of click-throughs received by the other two advertisers by assuming that all three received click-throughs in proportion to their respective CTR's for their positions (round to the nearest integer). If click-throughs remain, the publisher reallocates all three slots and starts a new phase.

If the phase ends because all click-throughs have been allocated, assume that the three advertisers received click-throughs in proportion to their respective CTR's (again, rounding if necessary).

Your task is to simulate the allocation of slots and to determine how many click-throughs each of the five advertisers get.

Answer:

Since in slot 1, the expected revenue for A is higher i.e., 0.0015

In slot2, C is selected because 0.00112 is higher

In slot3, E is selected because 0.0006 is higher

The first phase ends when A gets 10 Click-throughs and now A runs out of budget and it is not eligible for second phase.

B is selected to first slot because 0.014 is higher than C, D, E. Then C gets second slot and E gets third slot based on highest ones.

The second phase ends when B gets 22 Click-throughs and runs out of budget.

Now, in the third phase, C takes the first position and next second will be taken by D and third by E.

By ending of third phase, summing up Click-throughs for C, D, E in all phases, we get 36 click-throughs for C, 7 click-throughs for D, 26 click-throughs for E.

Click-throughs got for A are 10

Click-throughs got for B are 22

Click-throughs got for C are 36

Click-throughs got for D are 7

Click-throughs got for E are 26