Midterm Exam: Introduction to Big Data Analytics

Time: 1 hr. Total Marks: 60

Instructions (Oct 4, 2024):

1. **There are twelve questions. Each question is mandatory and carries five marks.**
2. **Answer options for each question are available on the Google Form. Multiple options may be correct.**
3. **Carefully check the question ID when attempting each question, as the IDs are intentionally created longer with repeating digits.**
4. **The backup option on the Google Form is disabled, and you can submit the form only once. Therefore, record your answer options in the answer copy as well. You should leave at least a 5-minute buffer at the end to enter your answers on the Google Form.**
5. **The IT division has been requested to log all access for registered students during the exam period. Please do not attempt to use unfair means. If caught cheating, institute policy will be strictly enforced.**
6. **No doubts will be entertained during the exam period, so do not approach any TA for clarification.**
7. **Write your complete calculations in the answer booklet. Simply selecting options in the Google Form may be awarded zero marks even if you have identified all the correct options .**
8. **Only completely correct answers will be awarded marks.**
9. **Do not write anything on the questions paper except your Roll number.**
10. **No extra answer sheets shall be provided.**
11. **All the best!**

Q: 111114083 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 we construct 3-bit sketches of the vectors by the random hyperplane method, using the randomly generated normal vectors r1, r2, and r3, in that order:
r1 = [1, -1, 1, -1, 1, -1]
r2 = [-1, -1, 1, 1, -1, 1]
r3 = [1, 1, 1, 1, 1, 1]

Construct the sketches of the three vectors u, v, w. Estimate the pairwise cosine similarities of u, v, and w from their 3-bit sketches. Which of the estimates corresponds to the computed 3-bit sketches?

Ans.

Take the dot product of each of u, v, and w with each of the normal vectors r1, r2, and r3. If the dot product is positive, the corresponding component of the sketch has a 1, and if it is negative, the sketch component is -1. The dot products and sketch are summarized in the table below.

r1	r2	r3	Sketch	
u	+1.25	-1.75	+1.75	+1, -1, +1
v	+0.95	-0.95	+1.35	+1, -1, +1
w	-0.35	+1.65	+1.85	-1, +1, +1

The estimate of the angle between two vectors is computed from their sketches, by multiplying 180 degrees by the fraction of the positions in which the two sketches DO NOT agree.

Since the sketches of u and v agree exactly, the estimate of the angle between them is 0, and the estimate of their cosine is therefore 1. Note that these vectors, while not identical are quite close.

The sketches of u and w agree in only one of the three components. Therefore, the estimate of their angle is (2/3)*180 = 120 degrees, and the estimate of their cosine is -0.5. The estimate for v and w is computed the same way that the estimate for u and w is computed.

Q: 111124083 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). Identify all the candidate similar pairs from the options.

Ans.

In the first band (first two rows) C1 and C4 both have (1,2), so they form a candidate pair. Also, C2 and C5 both have (2.3), so that is another candidate pair.

In the second band (rows 3 and 4) we find only C1 and C6 agree, and in the third band we find C1-C3 agree and C4-C7 agree. Thus, the five candidate pairs are C1-C4, C2-C5, C1-C6, C1-C3, and C4-C7.

Q: 111133431 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? Identify the correct choices from the options.

Ans. Here is the table of false positives and false negatives for various values of r and b:

r	b	False Positives	False Negatives
1	24	.99527	.00000059
2	12	.38729	.031676
3	8	.062236	.34361
4	6	.0095617	.67893
6	4	.0025598	.93895
8	3	.00000768	.98833
12	2	0	.99951
24	1	0	.99999

If M=N, we need to find the value of r than minimizes the sum of the last two columns; this value is r = 3, where it is .40585. If M=10N, then we need to minimize the sum of the last column (false negatives) and 10 times the third column (false positives). This minimum occurs at r = 4, where it is .77455. Similarly, the minimum of the last column plus 100 times the third column occurs at r = 5, where it is .96455. The minimum of the last column plus 1000 times the third column is at r = 6, where it is .99601.

Q: 111114099 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.

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. Then, our estimate will be the median of the three resulting values.

Identify from the options the set of three "random" timestamps that give the closest estimate to the true surprise number.

Ans.

First, the surprise number is 5*64 + 5*49 = 565. The reason is that the elements 1 through 5 appear 8 times, so they contribute $5*8^2$, and the elements 6 through 10 appear 7 times, contributing $5*7^2$.

Notice that for this contrived example, 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.

Q: 111122431 Suppose we are using the DGIM algorithm 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 the query asks us to estimate the number of 1's in the 21 most recent elements. What are the largest and smallest number of 1's that could be the correct answer, given the information above? What would the DGIM algorithm estimate? Identify all the true statements.

Ans. We know that each bucket ends with a 1, so we know that last position has a 1. we also know that, no matter how big the bucket is, it can only contribute a single 1 to the query window. Thus, we know exactly how many 1's are in the window suffix of length 21; it is 11.

However, DGIM would estimate the count as the sum of the sizes of all buckets included completely within the query window and half the next bucket, or 14.

Q: 111116139 Find the set of 2-shingles for the "document":

ABRACADABRA

and also for the "document":

BRICABRAC

Answer the following questions:

1. How many 2-shingles does ABRACADABRA have?
2. How many 2-shingles does BRICABRAC have?
3. How many 2-shingles do they have in common?

4. What is the Jaccard similarity between the two documents"?

Then, find all the true statements.

Ans.

The 2-shingles for ABRACADABRA: AB, BR, RA, AC, CA, AD, DA.

The 2-shingles for BRICABRAC: BR, RI, IC, CA, AB, RA, AC.

There are 5 shingles in common:AB, BR, RA, AC, CA.

As there are 9 different shingles in all, the Jaccard similarity is 5/9.

Q: 111103431 We wish to use the Flajolet-Martin algorithm 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 $3x + 7$ (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, identify all the set of four elements that gives the exactly correct estimate.

Ans. Here is a table of the hash values and resulting bit strings for each of the ten elements:

x	3x+7 (mod 11)	Bit String
1	10	1010
2	2	0010
3	5	0101
4	8	1000
5	0	0000
6	3	0011
7	6	0110
8	9	1001
9	1	0001
10	4	0100

In order to give the correct estimate (4), a set must have at most two 0's at the end of the hash value of any of its members, but must have a member whose hash has exactly two 0's at the end. Observe from the table above that 10 is the only element whose hash value has exactly two bits at the end. However, 1, 2, 3, 6, 7, 8, and 9 have zero or one 0 at the end, so the correct answers are any set of four elements that includes 10 and does not include 4 or 5.

Q: 111116334 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. Which of the options are the correct minhash value of the stated column? **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.

Ans. Look at the rows in the stated order R4, R6, R1, R3, R5, R2, and for each row, make that row be the minhash value of a column if the column has not yet been assigned a minhash value. We sart with R4, which only has 1 in column C3, so the minhash value for C3 is R4.

Next, we consider R6, which has 1 in C2 only. Since C2 does not yet have a minhash value, R6 becomes its value.

Next is R1, with 1's in C2 and C3. However, both these columns already have minhash values, so we do nothing.

Next, consider R3. It has 1's in C2 and C4. C2 already has a minhash value, but C4 does not. Thus, the minhash value of C4 is R3.

When we consider R5 next, we see it has 1's in C1 and C3. The latter already has a minhash value, but R5 becomes the minhash value for C1. Since all columns now have minhash values, we are done.

Q: 111128334 Suppose we use the two-stage algorithm to compute the product of matrices M and N. Let M have x rows and y columns, while N has y rows and z columns. As a function of x, y, and z, express the answers to the following questions:

1. The output of the first Map function has how many different keys? How many key-value pairs are there with each key? How many key-value pairs are there in all?
2. The output of the first Reduce function has how many keys? What is the length of the value (a list) associated with each key?
3. The output of the second Map function has how many different keys? How many key-value pairs are there with each key? How many key-value pairs are there in all?

Then, identify all the true statements.

Ans. Consider the first Map function. Each element of M is mapped to one pair, and this pair has a key equal to its column number. There are y different columns of M, and therefore that number of keys. Each column of M has x rows, and therefore, M is transformed into x different pairs for each key. Similarly, each element of N is mapped to a single pair, and the key of this pair is one of its y row numbers. Thus, there are only y different keys among all the pairs generated from M and N. Each row of N has z elements, so z pairs are produced from N for each key. We conclude that the output of the first Map function has y different keys, each key appears in x+z different pairs, and the total number of pairs is y(x+z).

The first Reduce function has input consisting of y elements. Each element is a key (i.e., a number that is a column of M and a row of N, associated with a list of values giving that column of M and that row of N. The Reduce function pairs the x elements of M on that list with the z elements of N on the same list, producing xz pairs, each with a key that is a coordinate (i,k) of the result matrix, and one element. Since there are y reducers, the total number of pairs produced is xyz, and each pair has a single value.

The second Map function is the identity, and so the output is the same xyz pairs that are input to the second Map stage.

Q: 111128749 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).

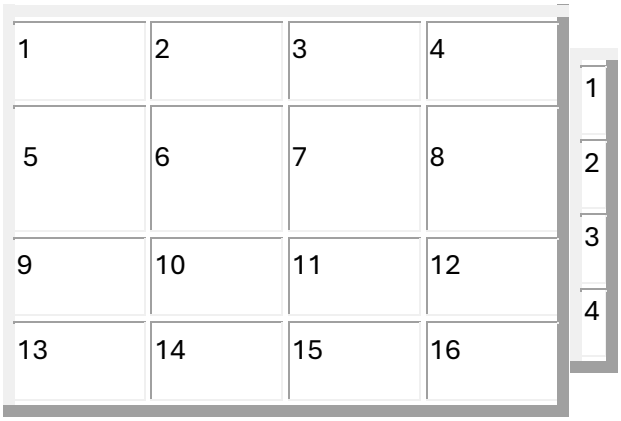
User ID's will be hashed to a bucket number, from 0 to 999,999. At all times, there will be a threshold t such that the 100-byte records for all the users whose ID's hash to t or less will be retained, and other users' records will not be retained. You may assume that each user generates emails at exactly 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? From the options, identify all the true

statements about a value of ***n*** and its value of ***t***. [Please note that (10)^11 = 10^{11}]

Ans. Suppose that the fraction of users in the sample is p. That is, 10^8p is the number of users whose records are stored. Since each user generates 10^{-8} of the emails in the stream, when n emails have been seen, the number of records stored is $10^8p10^{-8}n$ = pn. Note that is number does not depend on the number of users of the service.

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. That is, (t+1)/1,000,000 = $10^8/n$, or t = $10^{14}/n$ - 1.

Q: 111134099 Using the matrix-vector multiplication algorithm wherein each mapper has access to the complete vector (as discussed in the class), applied to the matrix and vector:



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1
2
3
4

apply the Map function to this matrix and vector. Then, identify all the key-value pairs that are output of Map.

Ans. Each m_{ij} is multiplied by v_j, and this product forms the value of a key-value pair that has key i, the row number. Thus, in row-major order, the sixteen key-value pairs produced are:

(1,1) (1,4) (1,9) (1,16)

(2,5) (2,12) (2,21) (2,32)

(3,9) (3,20) (3,33) (3,48)

(4,13) (4,28) (4,45) (4,64)

Q: 111134749 Suppose our input data to a mapReduce 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. Then, identify, in the list below, all the correct pairs that appear in the output.

Ans. Map does the following:
15 -> (3,15), (5,15)
21 -> (3,21), (7,21)
24 -> (2,24), (3,24)
30 -> (2.30), (3,30), (5,30)
49 -> (7.49)
We then group by keys, giving:

(2, [24, 30])
(3, [15, 21, 24, 30])
(5, [15, 30])
(7, [21, 49])

Finally, we add the elements of each list, giving the result (2,54), (3,90), (5,45), (7,70).