Data Mining and Machine Learning

Mid-Semester Examination, II Semester, 2023–2024

Date : 2 March, 2024	Marks : 30
Duration : 2 hours	Weightage : 20%

1. In the market-basket analysis problem, suppose the set of items I has size 10^7, the number of transactions T is 10^{10} and each transaction $t \in T$ contains at most 10 distinct items. Compute upper bounds for F_1 and F_2, the number of frequent itemset of size 1 and 2, respectively, for a support value of 0.1%. *(5 marks)*

2. Recall that a class association rule has the class attribute as its target. To reduce overfitting, a class association rule can be generalized by dropping attributes from its left hand side and checking if the performance improves over random test data.

   Given a decision tree, explain how to interpret paths in the tree as class association rules. How can we apply the generalization strategy for association rules to generalize decision trees? In what way would this be different from generalization through the usual method of pruning? *(5 marks)*

3. Your team has computed the solution to a linear regression problem on n attributes as $\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$. Your partner argues that the relative importance of the attributes can be computed from the coefficients. The most significant attribute is the one with the largest coefficient (in magnitude), the second most significant attribute is the one with the second largest coefficient, and so on. Explain whether your partner's claim is justified. *(5 marks)*

4. How can we use a random forest classifier to rank input features in order of importance? Why is this calculation more effective for a random forest than for a single decision tree? *(5 marks)*

5. We have a dataset $X = \{x_1, x_2, \ldots, x_N\}$ equipped with a symmetric distance function: $d(x_i, x_j) = d(x_j, x_i)$ is the distance between x_i and x_j. We construct an $N \times N$ matrix D such that $D[i,j] = d(x_i, x_j)$. We can cluster the N columns of D using the usual Euclidean distance in N dimensions, since each column is a vector of length N. Explain whether the clusters formed by the columns of D have any meaningful interpretation with respect to the original set X. *(5 marks)*

6. Explain how locally linear embeddings are computed. *(5 marks)*

- This exam has 4 questions for a total of 170 marks, of which you can score at most 100 marks.
- You may answer any subset of questions or parts of questions. All answers will be evaluated.
- Go through all the questions once before you start writing your answers.
- Use a pen to write. Answers written with a pencil will *not* be evaluated.
- Warning: CMI's academic policy regarding cheating applies to this exam.

You do *not* have to use loop invariants for proving the correctness of algorithms; but you must correctly explain why each loop (if there are some) does what you expect it to do. Of course, you *may* use loop invariants if you wish.

The arrays in this question paper are objects whose sizes are fixed, in the sense that the size cannot be changed after the array is created. In particular, these are *not* Python's lists, whose sizes can be changed using, say, append(). (Also: note that Python's list.append() does *not* run in *worst-case* constant time; be mindful of this when writing your pseudocode.)

Unstated assumptions and lack of clarity in solutions can and will be used against you during evaluation. You may freely refer to statements from the lectures in your arguments. You don't need to reprove these unless the question explicitly asks you to, but you must be precise.

Please ask the invigilators if you have questions about the questions.

1. Recall the SecondBest problem from Quiz 1:

SecondBest

- Input: An integer $n \geq 1$ and an array A of n integers. Array A is indexed from 0; its elements are thus $A[0], A[1], \ldots, A[(n-1)]$.
- Output: The *second largest* number x which is present in A, or the special value None if there is no such number in A.

Each part below shows (a Python version of) the pseudocode offered as a solution to this problem by someone among you. And in each case the pseudocode is *wrong*; it produces an incorrect output for certain valid inputs.

For each part, come up with a valid input of the form (n, A) where n is *at most* 5 which the given pseudocode fails to solve correctly. Clearly explain *how and why* the pseudocode fails to correctly solve this input.

You will get the credit for each part only if (i) the input that you specified is valid, (ii) the given pseudocode produces a wrong output for this input, (iii) you have described the actual output that the pseudocode produces when given this input, and, (iv) you have explained the reason why the pseudocode produces this wrong output.

Please make sure that you write the part number correctly, since there is no other way for me to match your answer to the part number in this question.

(a) [10]

```
def secondBest(n, A):
    if n == 1:
        return None
    else:
        x = A[0]
        y = None
    for i in range(1,n):
        if A[i] > x:
            y = x
            x = A[i]
    return y
```

(b) [10]

```
def secondBest(n, A):
    if n == 1:
        return None
    largest = A[0]
    secLargest = None
    for i in range(1,n):
        if A[i] < largest:
            secLargest = A[i]
    for i in range(1,n):
        if A[i] > largest:
            secLargest = largest
            largest = A[i]
    return secLargest
```

2. Consider the following problem:

Max Pair Product

- Input: An integer $n \geq 2$ and an array A of n non-negative integers. Array A is indexed from 0; its elements are thus $A[0], A[1], \ldots, A[(n-1)]$.
- Output: The *maximum value* of the product of two elements of A that are *at distinct indices* in A. Note that the two elements of A whose product is the required output, need not be distinct as numbers; but they must appear at different *indices* in A.

(a) Write the *complete* pseudocode for an algorithm MaxPairProduct(n, A) that solves the above problem in $\mathcal{O}(n)$ time and uses at most a constant amount of extra space, in the worst case. You will get the credit for this part only if your algorithm is correct and complete, and runs within the required time and space bounds. [15]

(b) Explain why your algorithm of part (a) is correct. [15]

(c) Prove that your algorithm from part (a) runs in $\mathcal{O}(n)$ time and constant extra space in the worst case. For this you may assume that each array operation, and each comparison of a pair of numbers, take constant time. Clearly state any other assumptions that you make. [10]

3. Assume for the sake of this question that 5000 candidates attempted Part A of CMI's 2023 MSc DS entrance exam. Consider the following claim and its proof: [10]

Claim

All the 5000 candidates who attempted part A of this exam, scored the exact same marks for part A.

Proof

By induction on the size of subsets of candidates who attempted Part A.

- Base case: Take any subset consisting of one candidate. Clearly, all candidates in this subset scored the same marks for part A (Because there is only one candidate in the subset.).
- Inductive assumption: suppose the claim holds for all subsets with at most k candidates.
- Inductive step: Let S be a subset with $k+1$ candidates. Remove an arbitrary candidate x from S to get the subset S' with k candidates. By the inductive assumption, all candidates in the set S' scored the same marks—say m'—for part A. Now remove another arbitrary candidate y ; $y \neq x$ from the *original* subset S to get a subset S'' with k candidates. By the inductive assumption, all candidates in the set S'' scored the same marks—say m''—for part A.

  Since the set S'' contains candidate x, we get that x scored the same marks—namely, m''—as every other candidate in the set $(S'' \setminus \{x\})$. Similarly, the set S' contains candidate y, and so we get that y scored the same marks—namely, m'—as every other candidate in the set $(S' \setminus \{y\})$.

  But notice that $(S'' \setminus \{x\}) = (S' \setminus \{y\})$. So we get that m' is in fact equal to m'', and that all the candidates in set S scored the same marks for part A.

Hence we get, by induction, that all the candidates scored the same marks.

It is obvious (I hope ...) that the claim cannot possibly be correct. Clearly explain what is wrong with the above proof.

✓4. Recall that a palindrome is a string that reads the same in either direction. A *non-trivial palindrome* is a palindrome with length (number of characters) at least two. Consider the following problem:

Palindrome Sequence

- Input: An integer $n \geq 2$ and a string S of length n. String S is an array indexed from 0; its elements are thus $S[0], S[1], \ldots, S[(n-1)]$.
- Output: True if S can be obtained by concatenating one or more non-trivial palindromes, and False otherwise. Equivalently: True if S can be partitioned (that is: cut up, without dropping any element) into one or more non-trivial palindromes, and False otherwise.

Some examples with $n = 10$:

- True instances: `nnllknkoyo`, `fjbubjfhuh`, `mttmzizzcz`, `fjfyspsyqq`, `abcdeedcba`
- False instances: `cmmcmhdaba`, `azatznnzth`, `ummnurlxmv`, `xqeppajynx`, `tyjglnvmaa`

(a) Write the *complete* pseudocode for a function IsNTP(w) that returns True if string w is a non-trivial palindrome and False otherwise, and runs in *linear time* in the length of w in the worst case. You will get the credit for this part only if your algorithm is correct and runs within the required worst-case time bound. [10]

(b) Write the *complete* pseudocode for a *recursive* function IsNTPSequence(n, S) that solves Palindrome Sequence. You may use the function IsNTP() that is described in part (a) as a black box even if you have not solved part (a). You will get the credit for this part only if your algorithm is (i) correct, and (ii) recursive. In particular, make sure that you have correctly handled all the base cases. [20]

~~(c)~~ Explain why your algorithm of part (b) correctly solves Palindrome Sequence. You may assume that the function IsNTP() works correctly. [10]

(d) Write a recurrence for the *number of recursive calls* that your algorithm from part (b) makes, in terms of n. Explain why your recurrence correctly captures this number. [10]

(e) Solve your recurrence of part (d) to get an upper bound on the number of recursive calls that your algorithm makes, in terms of n. [10]

(f) Write the *complete* pseudocode for a *memoized version* of your algorithm from part (b). As in part (b), you may use IsNTP() as a black box even if you haven't solved part (a). You will get the credit for this part only if your pseudocode is a correctly memoized version of your *correct* algorithm from part (b). Make sure that you have correctly handled all the sentinel values/base cases. [20]

(g) Show that your memoized algorithm of part (f) runs in $\mathcal{O}(n^c)$ worst-case time for input strings of length n, for some fixed constant c. What is the value of c that you get? [20]

Linear Algebra and its Applications
Mid-semester Examination
26/02/2024

(**Note:** You may use a standalone calculator, not your phone)

1. Find LU factorization of the following matrix: [10 points]

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

2. Given that the following matrix is symmetric, positive definite find its Cholesky factorization: [10 points]

$$A = \begin{bmatrix} 4 & 2 & 4 \\ 2 & 5 & 6 \\ 4 & 6 & 9 \end{bmatrix}$$

3. Let $w = [0, \dots, 0, w_{k+1}, \dots, w_n]^T$ be a column vector in $\mathbb{R}^n$ such that $1 \leq k \leq n$ and $w_{k+1} \neq 0$. Denote by e_k the k-th standard unit vector with 1 in k-th position and 0 everywhere else. Consider the following $n \times n$ matrix:

$$M_k^w := I_n - we_k^T.$$

Now answer the following questions: [10 points]

(a) Find the exact operation count needed to multiply two $n \times n$ matrices.

(b) Given an arbitrary $n \times n$ matrix C, find an algorithm to compute the product $M_k^w C$ whose operation count is significantly less (say, $O(n^2)$). Clearly write the algorithm (as a pseudocode) and show the operation count calculations.

4. Let $A = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$ be a 2×2 matrix such that ϵ is a very small real number. Answer the following questions: [15 marks]

(a) Find the condition number $\kappa_\infty(A)$ by explicitly calculating the inverse.

(b) Find the LU decomposition of A using GE *without pivoting.*

(c) What are the ∞-condition numbers of the factors L, U?

(d) Assume $1 - \epsilon^{-1} = -\epsilon^{-1}$ in U and find $\Delta A = LU - A$. What is the condition number of ΔA?

(e) Find the LU decomposition after permuting rows of A.

(f) Assume $1 - \epsilon = 1$ in U and find $\Delta A = LU - A$. What is the condition number ot ΔA?

(g) In which of the two scenarios above, the problem of solving $Ax = b$ (for any b and using GE, forward, backward substitutions etc.) is well-conditioned? In which method a small rounding error leads to a large backward error? Explain.

5. A floating-point number representation system is given as follows:

$$\mathcal{F} := \{0\} \cup \{\pm d_0.d_1d_2 \times 10^e \mid 1 \le d_0 \le 9; 0 \le d_1, d_2 \le 9; -9 \le e \le 9\}.$$

Now answer the following questions. [15 points]

(a) The number of normalized floating point numbers in $\mathcal{F}$ is:

(b) The smallest positive (nonzero) number in $\mathcal{F}$ is:

(c) The largest positive number in $\mathcal{F}$ is: 1.99×10^9

(d) The machine epsilon for $\mathcal{F}$ is: 9

(e) The relative error in representing 0.995 by an element in $\mathcal{F}$ is

(f) The smallest and largest possible gaps between any two consecutive elements of $\mathcal{F}$ are:

(g) The element of $\mathcal{F}$ that best represents the real number π is:

(h) The most accurate representation of

$$(1.23 \times 10^6) + (4.56 \times 10^4)$$

in $\mathcal{F}$ is:

(i) The most accurate representation of

$$(1.23 \times 10^1) \times (4.56 \times 10^2)$$

in $\mathcal{F}$ is:

(j) Let $x = 1.24 \times 10^1, y = 1.23 \times 10^0, z = 1.00 \times 10^{-3}$. Calculate the following:

$$\mathrm{fl}(\mathrm{fl}(x + y) + z).$$
$$\mathrm{fl}(x + \mathrm{fl}(y + z)).$$

CMI/2024/Mid-Semester Exam 01/03/2024

Distributed Computing and Big Data

Chennai Mathematical Institute

Duration: 90 Mins. Max: 25 Marks.

Instructions

- This is a closed book exam.
- This is an individual task. Do not discuss with anyone.
- No electronic devices are allowed. Wherever heavy calculation is involved, you need not evaluate it to the final number unless it is explicitly asked for. For example, it is acceptable to leave the answer as $\frac{1}{1+\frac{5}{32}}$. You need not evaluate it to 0.865.
- Clearly mention your name and roll number in your answer sheet.

Section 1: Correct answers carry 1 mark each. Answer True/False. Wrong answers carry -0.5 marks each.

Question 1. In the model of the distributed system as discussed in the class, there is no common global memory. [✓ True]/False?

Question 2. While computing average access time, head switching time is often considered negligible. [✓ True]/False?

Question 3. Data lakes are schemaless. [✓ True]/False?

✗ **Question 4.** Grid computing infrastructure refers to the use of heterogeneous systems. [✓ True]/False?

Question 5. Scalar time is strongly consistent. True/False?

Question 6. Hadoop uses a Write Once and Read Once model. True/False?

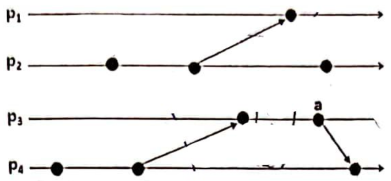
Question 7. Hadoop data nodes send periodic heartbeat signals and block reports to name node. [✓ True]/False?

Section 2: Correct answers carry 2 marks each. No negative marks.

✓ **Question 8.** Assume disk size = 512 GB, block size = 8 KB. How much space (in MB) will we need to store the free space bitmap?

✓ **Question 9.** As per Amdahl's law, What is the best achievable speed up if only 25% of the job can be parallelized, and we have 4 processors?

Question 10. If we were to annotate the following space-time execution diagram with vector time stamps, how would we annotate the event marked as 'a'?



Question 11. For the same diagram given above, annotate the events in p_3 with scalar time.

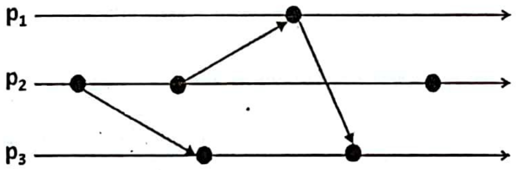
Question 12. For the same diagram given above, identify an inconsistent cut not involving the event marked as 'a' i.e., 'a' must be in the future of the cut.

Question 13. In the muddy children puzzle, as discussed in the class, what would the children say during the first and second rounds if n=4 and k=3? i.e., there are four children, and three of them have muddy forehead and to start with, they are told that at least one of them have muddy forehead. Assume that each child can only say 'yes', 'no' or 'dont know'. Use the following format to provide your answer.

```
Round1: <1st child response>,<2nd ...>,<3rd ...>,<4th ...>
Round2: <1st child response>,<2nd ...>,<3rd ...>,<4th ...>
```

Section 3: Question carries 3 marks. No negative marks.

Question 14. Annotate all the events in the following diagram with matrix time.



Question 15. You are given a large text file. You need to find all the distinct words that have no vowels. For example, if the input text file has the content, "my phone is ringing and ringing again", there are six distinct words (my, phone, is, ringing, and, again). The word without vowel is only one, "my". Therefore, the output should be a single word, "my".

Describe an approach using map reduce logic. No need to write any code.