CS 4650 Math-Programming Background Exam (Spring 2021)

Deadline: January 21st, 11:59pm ET

January 6, 2021

- Full Name:
- GT ID:
- GT E-mail:

Instruction

1. This background exam is designed to help students review the knowledge of linear algebra / probability / multivariate calculus, and programming skills. All of the questions represent materials that we would generally expect students to be familiar with before they take this class.
2. This background exam is to be completed alone. All work must be your own. Collaboration and discussion are all **NOT** allowed.
3. Submit your answers as a pdf file on Gradescope. We recommend students type answers with LaTeX or word processors. A scanned handwritten copy would also be acceptable. If writing by hand, write as clearly as possible. No credit may be given to unreadable handwriting. Each subproblem must be submitted on a separate page. When submitting to Gradescope (under Background Test), make sure to mark which page(s) correspond to each problem or subproblem. For instance, Q1 has 4 subproblems, so the solution to each must start on a new page.

   Note: This is a large class and Gradescope's assignment segmentation features are essential. Failure to follow these instructions may result in parts of your assignment not being graded. We will not entertain regrading requests for failure to follow instructions.
4. Write out all steps required to find the solutions so that partial credit may be awarded.

Q1. Linear Algebra	/8 pts
Q2. Probability	/6 pts
Q3. Multivariate Calculus	/8 pts
Q4. Programming (Algorithm)	/8 pts
Q5. Programming (NumPy)	/10 pts
Total	/40 pts

1 Linear Algebra

(a) (**3 pts**) Compute the l_1 norm, l_2 norm, and l_∞ norm of the vector $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

(b) (**2 pts**) $\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}$, $\mathbf{u} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Are $\mathbf{u}$ and $\mathbf{v}$ eigenvectors of $\mathbf{A}$?

(c) (**2 pts**) Provide answers to the following operations:

(i) $\begin{bmatrix} -3 & 5 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix}$

(ii) $\begin{bmatrix} -1 & 3 & 6 \\ 5 & 0 & 3 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 1 & -2 \\ 3 & 1 \end{bmatrix}$

(d) (**1 pts**) Compute vector $\mathbf{x}$ as the solution to the following linear equation: $\begin{bmatrix} 2 & 4 \\ 4 & 7 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 10 \\ 18 \end{bmatrix}$

2 Probability

(a) A probability density function is defined by

$$f(x) = \begin{cases} Ce^{-x} & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

(i) (**2 pts**) Compute C.

(ii) (**2 pts**) Compute the expected value of x, i.e. $E(x)$.

(b) (**2 pts**) Three locks are randomly matched with three corresponding keys. What is the probability that at least one lock is matched with the right key?

3 Multivariate Calculus

(a) (**2 pts**) $g(\mathbf{w}) = \mathbf{w}^T \mathbf{C} \mathbf{w} + \mathbf{b}^T \mathbf{w}$. Compute $\frac{\partial}{\partial \mathbf{w}} g(\mathbf{w})$.

(b) (**2 pts**) $f(x, y) = xy^2$. Compute $\int_0^1 \int_0^2 f(x, y)\,\mathrm{d}x\,\mathrm{d}y$.

(c) (**2 pts**) Find all the critical points of $f(x, y) = x^6 + y^3 + 6x - 12y + 7$.

(d) (**2 pts**) Calculate $\int_0^2 \int_x^2 e^{-y^2}\,\mathrm{d}y\,\mathrm{d}x$.

4 Programming (Algorithm)

(**8 pts**) Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. Fill in the program. The time complexity of your solution should be $O(mn)$, where m is the length of the first sequence X, n is the length of the second sequence Y.

```
def lcs(X: str, Y: str) -> int:
```

5 Programming (NumPy)

```
A = numpy.array([[1, 2], [3, 4]])
B = numpy.array([[4, 3], [2, 1]])
```

Compute:

(a) (**2 pts**) `A * B`, and `A.dot(B)`.

(b) (**2 pts**) `A > 2`, and `A < B`.

(c) (**3 pts**) `A.max(axis=0)`, and `B.min(axis=1)`, and `(A+B).sum()`.

(d) (**3 pts**) `A[0, 1] + B[1][0]`, and `A[:, 1] * B[1, 1]`, and `numpy.dot(A[:, 0], B[0, :])`.