New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 9223I: Midterm Exam.

Week of Oct. 26th, 2020, any 2 hour slot.

Directions

- *Absolutely no collaboration allowed.*
- You are allowed to use any notes or resources from the class, and any programming language, graphing software (e.g., Desmos), or symbolic math package (e.g., WolframAlpha).
- Show all of your work to receive full (and partial) credit.
- Clearly mark all submitted pages with the problem you are working on.

1. Always, sometimes, never. (**12pts – 3pts each**)

Indicate whether each of the following statements is **always** true, **sometimes** true, or **never** true. A correct answer receives full credit, but you can provide a short justification or example to explain your choice, which might earn partial credit if you are wrong.

(a) For random variables X and Y, $\mathbb{E}[X - Y] = \mathbb{E}[X] - \mathbb{E}[Y]$.
ALWAYS SOMETIMES NEVER

(b) For random variables X and Y, $\mathbb{E}[XY] \geq \mathbb{E}[X]\mathbb{E}[Y]$.
ALWAYS SOMETIMES NEVER

(c) For convex functions $f(x)$ and $g(x)$, $f(x) + g(x)$ is convex.
ALWAYS SOMETIMES NEVER

(d) For convex functions $f(x)$ and $g(x)$, $f(g(x))$ is convex.
ALWAYS SOMETIMES NEVER

2. Safety First (**5pts**)

An airplane has 1000 critical parts, including engine components, navigation equipment, etc. Each part has been thoroughly tested, and during a given flight, each part is guaranteed not to fail with probability 9999/10000. What is the probability that no part fails during a given flight? Give the highest bound you can based on the problem information.

3. Approximating the Trace of a Matrix (15pts)

Let $A \in \mathbb{R}^{n\times n}$ be a square matrix. Recall that the trace of A is the sum of its diagonal elements $(A) = \sum_{i=1}^{n} A_{ii}$.

class we saw that a matrix $\Pi \in \mathbb{R}^{m\times d}$ with **random Gaussian entries** satisfies the Johnson-Lindenstrauss Lemma with high probability when $m = O(\log n/\epsilon^2)$. Here we will consider the setting where Π's entries are **scaled random ± 1 random variables**. In particular, for all $i \in 1,\ldots,m$ and $j \in 1,\ldots d$, let

$$\Pi_{i,j} = \begin{cases} +\frac{1}{\sqrt{m}} & \text{with probability } 1/2. \\ -\frac{1}{\sqrt{m}} & \text{with probability } 1/2. \end{cases}$$

(a) (6pts) Let $\pi_i \in \mathbb{R}^d$ be the first row of Π. Show that for any $z \in \mathbb{R}^d$, $\mathbb{E}[\langle \pi_i, z\rangle^2] = \frac{1}{m}\|z\|_2^2$

(b) (6pts) Use (a) to conclude that for any two vectors $x, y \in \mathbb{R}^d$,

$$\mathbb{E}[\|\Pi x - \Pi y\|_2^2] = \|x - y\|_2^2$$

(c) (3pts) What's one reason you might you want to use a matrix with each entry equal to $\pm\frac{1}{\sqrt{m}}$ instead of being a random Gaussian?

4. Smoothness, strong convexity, and condition number. (10pts)

(a) (5pts) Draw below:

- A convex function which is smooth but not α-strongly convex for any $\alpha > 0$.
- A convex function which is strongly convex but not β-smooth for any finite β.
- A convex function which is not β-smooth for any finite β and not α-strongly convex for any $\alpha > 0$.

Extra credit (3pts): Given explicit expressions for your functions.

(b) (3pts) What is the condition number κ of the convex function $f(x) = x^2$. Recall that $\kappa = \frac{\beta}{\alpha}$ where β and α are the smoothness and strong convexity parameters of $f(x)$.

(c) (2pts) Does $f(x) = x^6$ have a finite condition number? Justify your answer.

5. Simple locality sensitive hash. (8pts)

Define the *hamming similarity* between two length d binary vectors $q, y \in \{0,1\}^d$ as:

$$1 - \frac{\|q-y\|_1}{d}.$$

Here $\|q-y\|_1$ is the ℓ_1 distance, which is defined as $\|z\|_1 = \sum_{i=1}^d |z_i|$.

(a) (4pts) Let g be a uniform random integer in $\{1, \ldots, d\}$. Define the function $h : \{0,1\}^d \to \{0,1\}$ as $h(x) = x_g$, where x_g is the g^{th} entry in the vector x. Show that h is a locality sensitive hash function for hamming similarity.

(b) (4pts) What are **two reasons** to *not use* locality sensitive hashing for similarity search, but instead to perform a linear scan.