New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 9223I: Midterm Exam, Solutions

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

Indicate whether each of the following statements is **always** true, **sometimes** true, or **never** true. Provide a short justification or example to explain your choice.

(a) For random variables X and Y, $\mathbb{E}[X - Y] = \mathbb{E}[X] - \mathbb{E}[Y]$.

ALWAYS SOMETIMES NEVER

Follows from linearity of expectation, which holds for any r.v.'s.

(b) For random variables X and Y, $\mathbb{E}[XY] \geq \mathbb{E}[X]\mathbb{E}[Y]$.

ALWAYS **SOMETIMES** NEVER

Let X be uniform random ± 1. If $Y = X$ we have $1 = \mathbb{E}[X]\mathbb{E}[Y] \geq \mathbb{E}[X]\mathbb{E}[Y] = 0$. If $Y = -X$ we have $-1 = \mathbb{E}[X]\mathbb{E}[Y] < \mathbb{E}[X]\mathbb{E}[Y] = 0$.

(c) For convex functions $f(x)$ and $g(x)$, $f(x) + g(x)$ is convex.

ALWAYS SOMETIMES NEVER

Follows from first definition of convexity. Let $h(x) = f(x) + g(x)$. For any $\lambda \in [0, 1]$, $h(\lambda x + (1-\lambda)y) = f(\lambda x + (1-\lambda)y) + g(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y) + \lambda g(x) + (1-\lambda)g(y) = \lambda h(x) + (1-\lambda)h(y)$.

(d) For convex functions $f(x)$ and $g(x)$, $f(g(x))$ is convex.

ALWAYS **SOMETIMES** NEVER

Both $g(x) = x$ and $g(x) = -x$ are convex. $f(g(x))$ is convex for the former, but not for the later (in fact, it's concave).

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.

We cannot assume that failures are *independent*. The best bound we can give is via a union bound:

$$\Pr[\text{no failures}] = 1 - \Pr[\text{at least on failure}] \geq 1 - \sum_{i=1}^{1000} \Pr[\text{part } i \text{ fails}] \geq 1 - 1000 \cdot (1 - 9999/10000) = 9/10.$$

So the best upper bound we can give is **9/10**.

3. Johnson-Lindenstrauss with Sign Matrices (15pts)

In 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 $z \in \mathbb{R}^d$, $\mathbb{E}[\langle \pi_i, z\rangle^2] = \frac{1}{m}\|z\|_2^2$.

Let $W = \langle \pi_i, z\rangle$. W is a random variable and we want to given an expression for $\mathbb{E}[W^2]$. Let $X_1, \ldots, X_d$ be independent random variables that take values ± 1, each with probability $1/2$. Then observe that:

$$W = \frac{1}{\sqrt{m}}\sum_{i=1}^{d} X_i z_i.$$

By linearity of expectation $\mathbb{E}[W] = 0$ since $\mathbb{E}[X_i z_i] = 0$ for all i. $\mathrm{Var}[W] = \frac{1}{m}\mathrm{Var}[\sum_{i=1}^d X_i z_i]$ and, since all $X_i z_i, X_i z_j$ are independent, we can apply linearity of variance.

$$\mathrm{Var}[\sum_{i=1}^{d} X_i z_i] = \sum_{i=1}^{d}\mathrm{Var}[X_i z_i] = \sum_{i=1}^{d} z_i^2 = \|z\|_2^2.$$

We conclude by noting that,

$$\mathbb{E}[W^2] = \mathbb{E}[W^2] - 0 = \mathbb{E}[W^2] - \mathbb{E}[W]^2 = \mathrm{Var}[W] = \frac{1}{m}\|z\|_2^2.$$

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

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

First note that:

$$\|\Pi z\|_2^2 = \sum_{i=1}^{m}\langle \pi_i, z\rangle^2$$

and so by linearity of expectation, $\mathbb{E}\|\Pi z\|_2^2 = \sum_{i=1}^m \mathbb{E}\langle \pi_i, z\rangle^2 = \|z\|_2$.

Plugging in $z = x - y$ and noting that $\Pi(x - y) = \Pi x - \Pi y$ gives the result.

(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?

- Less storage space for Π (1 bit per entry instead of a floating point number).
- Faster time to compute Πx: only additions/subtractions, no floating point multiplications.
- Faster to generate Π (how to generate a random Gaussian isn't obvious).

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.

- $f(x) = x$
- $f(x) = x^6$ (see below). There are simpler examples to draw too.
- $f(x) = |x|$.

(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 $f(x)$.

$\nabla f(x) = 2x$. For all x, y we have,

$$|\nabla f(x) - \nabla f(y)| = 2|x - y|$$

so from the definition of β smooth, $f(x)$ is 2-smooth.

We also have:

$$\nabla f(x)(x-y) - [f(x) - f(y) = 2x^2 - 2xy - x^2 + y^2 = (x-y)^2$$

So, by the definition of α smoothness, $f(x)$ is 2-smooth. We conclude that $\boldsymbol{\kappa} = \mathbf{1}$ for $f(x) = x^2$.

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

It does not because $f(x)$ is not β-smooth for any finite β. In particular, $|\nabla f(x) - \nabla f(y)| = 6|x^5 - y^5|$. Setting $y = 0$, this can be arbitrarily larger than $|x - y|$ as $x \to \infty$

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.

Note that for binary vectors x, y, $\|x - y\|_1$ is exactly the number of entries where the vectors *differ*. $d - \|x - y\|_1$ is the number of entries where the vectors are the same. So we have, for any binary x, y,

$$\Pr[h(x) = h(y)] = \Pr[x_g = y_g] = \frac{d - \|x-y\|_1}{d} = 1 - \frac{\|q-y\|_1}{d}.$$

So our collision probability is *exactly proportional* to the hamming similarity. It increases when similarity increases, and decreases when it decreases. So h is a locality sensitive hash function.

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

- LSH requires preprocessing time to hash all database elements into multiple tables. This is slower than a linear scan. So you only save time with LSH if you have many queries.
- LSH requires more space.
- LSH always fails with some probability – it can never be 100% reliable.