COMPSCI 514 Spring 2026: 1 Midterm Review

1 Logistics

- The exam is closed book, no calculator or cheatsheets allowed.
- Held **in class** Thursday 3/12.

2 Studying Suggestions

- Focus primarily on practice questions. First from the past midterms, then from this sheet/questions I have asked in class, then from the textbooks if needed.
- Make sure you understand all the homework and quiz problem solutions.
- Review slides as needed to fill in gaps in knowledge, but don't make this your primary means of studying.

3 Concepts to Study

Foundational Probability Concepts + Concentration Bounds

- Linearity of expectation and variance.
- Markov's inequality, Chebyshev's inequality – should be able to apply and also understand how they were derived.
- Union bound.
- General idea of higher moment inequalities. But do not need to know in detail. Don't need to be able to apply moment generating function.
- Chernoff and Bernstein bounds – should be able to recognize when they can be applied (i.e., require a sum of bounded independent random variables) and apply them. Do not need to know how they are derived (we didn't cover in detail), or memorize the formulas. I will provide the formulas on the exam if they are needed.
- General idea of law of large numbers and central limit theorem. Connection to concentration inequalities (i.e., Chebyshev's inequality $\rightarrow$ law of large numbers, exponential concentration bounds $\rightarrow$ central limit theorem.)
- Technique of breaking random variables into sums of indicator random variables.
- Bounds based on 'collision counting' (see e.g., Captcha example, two level hashing, etc.)

- Averaging to reduce error.
- Median trick and why it can be more robust than just averaging. How to analyze it.

Random Hashing and Related Algorithms

- Random hash functions.
- Definitions of 2-universal and pairwise independent hash functions. Why we care about these types of hash functions.
- 2-level hashing.
- Application of random hashing to load balancing.
- Bloom Filters, their space complexity and guarantees, how the algorithm works. Should be able to apply the false positive rate equation, but do not need to memorize it.
- MinHash for Jaccard similarity.
- What a locality sensitive hash (LSH) function is.
- How LSH is used for similarity search (with hash signatures and repeated tables).
- Idea of s-curve and what it tells us. How the s-curve changes as we vary signature length r and table count t.

Streaming Algorithms

- Frequent elements problem definition and setup.
- Count-min sketch and analysis. Should understand this in detail.
- Min-Hashing for Distinct Elements. Understand the 'idealized' algorithm where we hash to real numbers. Understand its analysis.

4 Practice Questions

Probability, Expectation, Variance:

1. Prove that for any random variable $\mathbf{X}$, $Var[\mathbf{X}] = \mathbb{E}[\mathbf{X}^2] - \mathbb{E}[\mathbf{X}]^2$.
2. Show that for any $\mathbf{X}$, $\mathbb{E}[\mathbf{X}^2] \geq \mathbb{E}[\mathbf{X}]^2$. (Hint: use the identity proven above).
3. Show that for independent $\mathbf{X}$ and $\mathbf{Y}$, $\mathbb{E}[\mathbf{X} \cdot \mathbf{Y}] = \mathbb{E}[\mathbf{X}] \cdot \mathbb{E}[\mathbf{Y}]$.
4. Show that for independent $\mathbf{X}$ and $\mathbf{Y}$ with $\mathbb{E}[\mathbf{X}] = \mathbb{E}[\mathbf{Y}] = 0$, $Var[\mathbf{X} \cdot \mathbf{Y}] = Var[\mathbf{X}] \cdot Var[\mathbf{Y}]$. **Hint:** use part (3).
5. Given a random variable $\mathbf{X}$, can we conclude that $\mathbb{E}[1/\mathbf{X}] = 1/\mathbb{E}[\mathbf{X}]$? If so, prove this. If not, give an example where the equality does not hold.
6. The maximum score on a midterm exam is 100% and the class average is 75%. What's the maximum fraction of students who could have scored below 25%.

7. For the statements below, indicate if they are **always true**, **sometimes true**, or **never true**. Give a sentence explaining why.

   (a) $\Pr[\mathbf{X} = s \cap \mathbf{Y} = t] > \Pr[\mathbf{X} = s]$. ALWAYS SOMETIMES NEVER

   (b) $\Pr[\mathbf{X} = s \cup \mathbf{Y} = t] \leq \Pr[\mathbf{X} = s] + \Pr[\mathbf{Y} = t]$. ALWAYS SOMETIMES NEVER

   (c) $\Pr[\mathbf{X} = s \cap \mathbf{Y} = t] = \Pr[\mathbf{X} = s] \cdot \Pr[\mathbf{Y} = t]$. ALWAYS SOMETIMES NEVER

8. Exercises 2.1, 2.4, 2.6, 2.8 (spherical Gaussian means entries are independent), 2.9 (challenging but interesting), 2.41, and 6.10 of *Foundations of Data Science* (https://www.cs.cornell.edu/jeh/book.pdf).

Concentration Inequalities:

1. Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be the number of visitors to a website on n consecutive days. These are independent and identically distributed random variables. For every i, we have $\mathbb{E}[\mathbf{X}_i] = 20,000$ and $Var[\mathbf{X}_i] = 100,000,000$.

   (a) Give an upper bound on the probability that on day i, more than $40,000$ visitors hit the website.

   (b) Let $\bar{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$ be the average number of visitors over n days. What are $\mathbb{E}[\bar{\mathbf{X}}]$ and $Var[\bar{\mathbf{X}}]$?

   (c) Give an upper bound on the probability that $\bar{\mathbf{X}} \geq 25,000$, for $n = 100$. (Which concentration bounds can you use here? 2/4 that we learned in class will apply. Think about why the other two do not.)

   (d) In reality do you expect $\mathbf{X}_1, \ldots, \mathbf{X}_n$ to be independent and identically distributed?

2. Assume there are 1000 registered users on your site $u_1, \ldots, u_{1000}$, and in a given day, each user visits the site with some probability p_i. The event that any user visits the site is independent of what the other users do. Assume that $\sum_{i=1}^{1000} p_i = 500$.

   (a) Let $\mathbf{X}$ be the number of users that visit the site on the given day. What is $\mathbb{E}[\mathbf{X}]$.

   (b) Apply a Chernoff bound to show that $\Pr[\mathbf{X} \geq 600] \leq .01$.

   (c) Apply Markov's inequality and Chebyshev's inequality to bound the same probability. How do they compare?

3. Give an example of a random variable and a deviation t where Markov's inequality gives a tighter bound than Chebyshev's inequality.

4. Which of these inequalities can you use to bound the a sum independent normal random variables: Markov's, Chebyshev's, Bernstein, Chernoff. What about the sum of independent rolls of a 6-sided dice?

5. Why can you not use Bernstein's inequality to bound a some of independent Chi-Squared random variables as we do (via other means) in the distributional JL lemma proof?

6. Let $\mathbf{X}$ be the sum of n independent 6-sided dice rolls. What is $\mathbb{E}[\mathbf{X}]$? What is $Var[\mathbf{X}]$?. Use Chebyshev's inequality to show that $\Pr[|\mathbf{X} - \mathbb{E}[\mathbf{X}]| \geq s\sqrt{n}] = O(1/s^2)$. Use Bernstein's inequality to show that $\Pr[|\mathbf{X} - \mathbb{E}[\mathbf{X}]| \geq s\sqrt{n}] = O(e^{-cs})$ for some constant c. Which of these bounds is stronger?

7. Consider sampling n independent random variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$ each with variance $Var(\mathbf{X}_i) = \sigma^2$ and mean $\mathbb{E}[\mathbf{X}_i] = \mu$. Let $\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$ be the sample mean.

   (a) What is $Var(\hat{\boldsymbol{\mu}})$?

   (b) Give an upper bound on the probability that $\hat{\boldsymbol{\mu}}$ falls t standard deviations from the true mean. I.e., on $\Pr[|\hat{\boldsymbol{\mu}} - \mu| \geq t \cdot \sigma]$.

   (c) Given parameters $\epsilon, \delta \in (0, 1)$, how many samples n must you take so that $|\hat{\boldsymbol{\mu}} - \mu| \leq \epsilon \cdot \sigma$ with probability at least $1 - \delta$?

   (d) Describe in a sentence or two how parts (a)-(c) relate to the law of large numbers.

   (e) You would like to prove a tighter bound than part (b) using Bernstein's inequality or a Chernoff Bound. What additional assumptions on $\mathbf{X}_1, \ldots, \mathbf{X}_n$ would you need if you wanted to apply each of these concentration inequalities?

   (f) Without such assumptions, what estimation strategy can you use to improve the dependence on $1/\delta$ in the sample complexity bound of part (c)?

Bloom Filters:

1. Bloom Filters allow you to store a set of n items with small false positive query rate in $o(n)$ space. ALWAYS SOMETIMES NEVER

2. Consider storing a set of m items in a Bloom Filter. If I want to achieve false positive rate $\delta \in (0, 1)$, show that the filter requires $O(m \log(1/\delta))$ space and $O(\log 1/\delta)$ query time.

3. Consider a bloom filter B_S storing set S and another bloom filter B_R storing set R. Assume that both bloom filters were constructed using the same hash functions. How can we construct a bloom filter $B_{S \cup R}$, which stores the union of the two sets, without accessing the items of the sets? What is the runtime to construct this filter?

4. Consider a Bloom filter storing m items, using $16m$ bits of space, and using the optimal number of hash functions (don't worry about rounding this to an integer). What is the false positive rate δ? What is the false positive rate of a Bloom filter that stores $m/2$ items in $8m$ bits of space and uses the optimal number of hash functions? What about one that stores $m/2$ items in $16m$ bits of space.

General Random Hashing and Streaming Algorithms:

1. Use a Chernoff bound to show that if we hash n items into a table with n buckets, with probability $\geq 1 - \delta$, the maximum number of items in a single bucket is upper bounded by $O(\log(n/\delta))$. What bound do you get if you apply Chebyshev's inequality instead?

2. Consider an algorithm $\mathcal{A}$ running in time $T(\mathcal{A})$, that with probability .6 outputs an estimate of the number of triangles in an input graph up to error ± 100, and with probability .4 outputs some bad estimate with worse error. Describe an algorithm that outputs an estimate of the number of triangles in an input graph up to error ± 100 with probability $\geq .99$ and runs in time $O(T(\mathcal{A}))$.

3. Let $\mathbf{X} = max\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d\}$, where the $\mathbf{X}_i$ are distributed independently and uniformly at random in $[0, 1]$. What is $\mathbb{E}[\mathbf{X}]$? What is $Var(\mathbf{X})$? Describe an algorithm that takes $O(\frac{1}{\epsilon^2 \cdot \delta})$ independent samples of $\mathbf{X}$ and estimates d up to error $\pm \epsilon d$ with probability at least $1 - \delta$.

4. We showed that if you implement count-min sketch with $m = O\left(\frac{k}{\epsilon}\right)$ length arrays and $t = O(\log(1/\delta))$ repetitions, for every x appearing in your data stream, with probability at least $1 - \delta$,
$$f(x) \leq \tilde{f}(x) \leq f(x) + \frac{\epsilon n}{k}.$$
   (a) How many repetitions t are required if you want this guarantee to hold simultaneously for all n items in your datastream, with probability at least $1 - \delta$?
   (b) How many repetitions are required if you want this guarantee to hold simultaneously for all n items in your datastream *and* at any time step. That is, let $f_i(x)$ be the number of times that x occurs in the first i steps (so $f(x) = f_n(x)$). Let $\tilde{f}_i(x)$ be the count-min sketch estimate at the i^{th} step. At all $i \leq n$, you want your estimate for every item to satisfy: $f_i(x) \leq \tilde{f}_i(x) \leq f_i(x) + \frac{\epsilon i}{k}$.
   (c) At each time step i of the algorithm you store a list of any item with estimated frequency $\tilde{f}_i(x) \geq \frac{i}{k}$. Show that for $\epsilon < 1/2$, assuming the guarantee of part (b) holds, this list never contains more than $O(k)$ items. Argue that at the end of the stream it contains all items with frequency $\geq \frac{n}{k}$ and none with frequency $\leq (1 - \epsilon) \cdot \frac{n}{k}$.

5. Exercises 6.1, 6.2, 6.6, 6.7, 6.19, 6.21 (challenging but interesting problem on shingling) 6.22, 6.23 of *Foundations of Data Science*

Locality Sensitive Hashing:

1. Why would you use locality sensitive hashing over regular hash tables with a random hash function?

2. A pairwise independent hash function is locality sensitive. ALWAYS SOMETIMES NEVER

3. Consider a hash function mapping m-bit strings to a single bit – $\mathbf{h} : \{0,1\}^m \to \{0,1\}$. We generate $\mathbf{h}$ by selecting a random position i from $1, \ldots, m$. Then let $\mathbf{h}(x) = x(i)$, the value of x at position i. Note that after i is chosen, it remains fixed, when we apply $\mathbf{h}$ to different inputs.
   (a) Given $x, y \in \{0,1\}^m$ with hamming distance $\|x - y\|_0$ (i.e., x and y have different bit values in $\|x - y\|_0$ positions), what is $\Pr[\mathbf{h}(x) = \mathbf{h}(y)]$.
   (b) Is $\mathbf{h}$ a locality sensitive hash function?
   (c) Let m be the number of all possible 5-singles in a document (i.e., all possible strings of 5 English words). If x and y are indicator vectors of the 5-shingles in two different documents, why do we expect them to be very sparse (i.e., each only have a few bits set to 1)? **Hint:** What is the maximum number of 5 shingles that a single document with w words in it can contain.
   (d) Why might MinHash and Jaccard similarity be more useful in the situation of (c) than the hash function $\mathbf{h}$ and Hamming distance.

4. You would like to use shingling and locality sensitive hashing to identify possible plagiarism in student essays. One possibility is to compare an essay A with a publication B in your database using shingling and Jaccard similarity. Another possibility is to use shingling, but then to measure similarity with cosine similarity, where the shingle sets are viewed as binary vectors. In the following questions, consider an essay A with 1000 unique length-c shingles in it and a publication B with 3000 unique length-c shingles, let S_A and S_B be the shingle sets

for A and B respectively. So $|S_A| = 1000$ and $|S_B| = 3000$. Assume the essay A was fully copied from a portion of B. Note that the shingle size c is arbitrary and will not factor into any of the solutions below.

Note: Some of the calculations in this problem are beyond what you would see on the exam, as you cannot use a calculator and so won't have to do any heavy computation. But worthwhile working through to give an understanding of s-curve tuning for LSH and the different between different similarity metrics and hash functions.

(a) What is the Jaccard similarity between the shingle sets of the two documents? What is $\Pr(MinHash(S_A) = MinHash(S_B))$?

(b) Let m be the number of all possible length-c shingles (i.e., all ordered sets of c words in the English language). Represent S_A and S_B by length m binary vectors x_A and x_B, with a 1 in every position corresponding to a shingle they contain. What is the cosine similarity between x_A and x_B (again assuming A was fully copied from a portion B)? What is $\Pr(SimHash(x_A) = SimHash(x_B))$? Use that for any two vectors z, w, $\Pr(SimHash(z) = SimHash(w)) = 1 - \frac{\theta}{\pi}$ where θ is the angle between z and w in radians.

(c) Consider another essay C also with 1000 unique shingles that is not copied and only shares 100 shingles with B. Compute $\Pr(MinHash(S_C) = MinHash(S_B))$. Compute $\Pr(SimHash(x_C) = SimHash(x_B))$.

(d) For both MinHash and SimHash, find a signature length r and repetition parameter t such that the fully copied essay A is identified with LSH-based similarity search with probability $\geq .95$ and the non-copied essay C is identified with probability $\leq .05$. Focus on minimizing the space complexity (i.e., the number of hash tables t used). By 'identified', we mean that the essay falls in the same bucket as B in at least one of the t hash tables. **Note:** It may be helpful to write a very simple program to help solve this one.

(e) Given the above, which similarity metric and hash function would you pick for the plagiarism detection task?