COMPSCI 514 Fall 2026: Midterm 2 Review

1 Logistics

- The exam is closed book, no calculator or cheatsheets allowed.
- Held **in class Wednesday 11/18**.
- Format will be similar to Midterm 1: TRUE/FALSE questions, numerical answer questions (both with explanations required), along with two longer 'homework style' questions.

2 Studying Suggestions

- Focus primarily on practice questions. First from the past Midterm 1s/Midterm 2s/Finals, 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

Probability and Randomized Algorithms (Pre-Midterm 1)

- The exam will not specifically test material covered before Midterm 1, but you will need to be able to apply foundational techniques. E.g., compute expectations, linearity of expectation, union bound, take the expectation of a random matrix or a random dot product, etc.

Locality Sensitive Hashing and Nearest Neighbor Search

- What a locality sensitive hash (LSH) function is.
- MinHash for Jaccard similarity.
- SimHash for Cosine similarity.
- How locality sensitive hashing (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.
- Definition of greedy search and navigable graphs.

- Application to approximate nearest neighbor algorithms.
- Do not need to be able to full recreate the $O(\sqrt{n \log n})$ degree navigable graph construction from lecture, but should understand the ideas behind it.

Low Distortion Embedding and JL Lemma

- Low-distortion embeddings definition.
- The Johnson-Lindenstrauss Lemma, and how to apply it.
- The distributional Johnson-Lindenstrauss Lemma and its connection to the standard Johnson-Lindenstrauss Lemma.
- Do not need to be able to full recreate the JL proof, but should understand the ideas behind it and be comfortable thinking about random vectors and matrices as we do in the proof.
- Vector quantization and motivation.
- Quantized JL algorithm for vector quantization. You should know what the algorithm is, and understand the key analysis ideas, but don't need to fully recreate the proof.
- How JL, QJL, and SimHash all related to each other (i.e., not quantizing at all, quantizing one vector, quantizing both vectors).

Low-Rank Approximation and PCA

- Understand and apply basic linear algebraic manipulations used. E.g.:
  - $y^T y = \|y\|_2^2$ and using this to split $\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^T y$.
  - $\text{tr}(\mathbf{A}\mathbf{A}^T) = \text{tr}(\mathbf{A}^T\mathbf{A}) = \sum_{i=1}^{d} \lambda_i(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}\|_F^2$
  - For $\mathbf{V} \in \mathbb{R}^{d \times k}$ with orthonormal columns, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T$ is a projection matrix.
  - By Pythagorean theorem $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$.
  - Definition of eigenvectors and values.
  - Courant-Fischer theorem and connection to low-rank approxmiation.
- Low-rank approximation as projection onto a k-dimensional subspace. How this projection gives a compressed representation of a data matrix $\mathbf{X}$.
- Dual view of low-rank approximation as finding k vectors that approximately span the rows (data points) and the columns (features). High level understanding of why a data matrix may be nearly low-rank.
- Finding the best low-rank approximation (i.e., the best orthonormal span $\mathbf{V} \in \mathbb{R}^{d \times k}$) of $\mathbf{X}$ using the eigenvectors of $\mathbf{X}^T\mathbf{X}$. Do not need to have full derivation memorized, but it is worth working through. Understand high level takeaways.
- Measuring the quality of the optimal low-rank approximation by plotting the eigenvalues (the spectrum).
- Ability to recognize when a matrix will be low-rank or close to low-rank (e.g., given in image, be able to make an educated guess about what its spectrum looks like.)

4 Practice Questions

1. Locality Sensitive Hashing and Nearest Neighbor Search

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?

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) Given an essay A with 1000 words in it an a book B with 10000 words in it, what is the maximum number of 7 word shingles that may appear in A? What is the maximum that may appear in B?

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

   (c) 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.

   (d) 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 $\Pr(SimHash(x_C) = SimHash(x_B))$.

(e) 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.

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

2. Low Distortion Embeddings and the JL Lemma

1. Let $\vec{g} \in \mathbb{R}^d$ have i.i.d. $\mathcal{N}(0,1)$ entries. Let $\vec{v} \in \mathbb{R}^d$ be fixed. What are $\mathbb{E}[\langle \vec{g}, \vec{v} \rangle]$ and $\text{Var}[\langle \vec{g}, \vec{v} \rangle]$. How is $\langle \vec{g}, \vec{v} \rangle$ distributed? What about $\langle \vec{g}, \vec{v} \rangle^2$? What about $\text{sign}(\langle \vec{g}, \vec{v} \rangle)$?

2. Let $\vec{g} \in \mathbb{R}^d$ have i.i.d. entries chosen uniformly in $\{-1, 1\}$. Let $\vec{v} \in \mathbb{R}^d$ be fixed. What are $\mathbb{E}[\langle \vec{g}, \vec{v} \rangle]$ and $\text{Var}[\langle \vec{g}, \vec{v} \rangle]$. What about $\langle \vec{g}, \vec{v} \rangle^2$? What about $\text{sign}(\langle \vec{g}, \vec{v} \rangle)$? Why might a random $\vec{g}$ of this form be preferable to random Gaussian $\vec{g}$ in practice.

3. Consider unit vectors $x_1, \ldots, x_n \in \mathbb{R}^d$. Prove that if $\Pi \in \mathbb{R}^{m \times d}$ is a random Gaussian matrix with $m = O(\log(n/\delta)/\epsilon^2)$ then with probability at least $1 - \delta$, for all i, j

$$|\langle \Pi x, \Pi y \rangle - \langle x, y \rangle| \leq \epsilon.$$

Hint: Apply the distributional JL lemma to conclude that for all i, $\|\Pi x_i\|_2^2 \in (1 \pm \epsilon)$ and for all i, j, $\|\Pi x_i - \Pi x_j\|_2^2 \in (1 \pm \epsilon)\|x_i - x_j\|_2^2$. Use these two bounds to conclude the bound on the inner product.

4. How does the above bound compare to the bound we would get from applying quantized JL (QJL)? If we analyze QJL with a Chebyshev's inequality and a union bound, we will have a very bad dependence on n in the bound. How can get improve this?

5. Why is Johnson-Lindenstrauss compression applied for vector quantization, as opposed to PCA?

3. Linear Algebra and Low-Rank Approximation

1. Exercises 3.6, 3.7, 3.8, 3.10, 3.11 (here $|\vec{x}|$ denotes the Euclidean norm of $\vec{x}$), 3.12, 3.13, 3.15, 3.18, 3.20, 3.21, 3.22, 3.26, 7.16, 12.31, 12.33 *Foundations of Data Science.*

2. Linear algebra practice:

   (a) For any vector $\vec{y}$ verify that $\|\vec{y}\|_2^2 = \langle \vec{y}, \vec{y} \rangle = \vec{y}^T \vec{y}$.

   (b) If $\mathbf{X} = \mathbf{AB}$, $\mathbf{X}$'s columns are spanned by the columns of $\mathbf{A}$ and $\mathbf{X}$'s rows are spanned by the rows of $\mathbf{B}$. Check that you understand why. What about when $\mathbf{X} = \mathbf{ABC}$ for some matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$. If $\text{rank}(\mathbf{A}) = k$, prove that $\text{rank}(\mathbf{X}) \leq k$.

   (c) For $\mathbf{V} \in \mathbb{R}^{n \times k}$ with orthonormal columns and vector $x \in \mathbb{R}^n$, $\|\mathbf{V}^T x\|_2 = \|x\|_2$. Always? Sometimes? Never?

   (d) Letting $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ have columns equal to the top k left singular vectors of $\mathbf{X}$ and $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ have columns equal to the top k right singular vectors of $\mathbf{X}$, $\mathbf{U}_k \mathbf{U}_k^T \mathbf{X} = \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T$. Always? Sometimes? Never?

(e) Prove the matrix Pythagorean theorem: that if $\mathbf{V} \in \mathbb{R}^{d \times k}$ has orthonormal columns, then for any matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$.

(f) For any $\mathbf{V} \in \mathbb{R}^{d \times k}$ with orthonormal columns, $\mathbf{V}\mathbf{V}^T$ is the projection matrix onto the subspace spanned by the columns of $\mathbf{V}$ ($\mathbf{V}$'s column span). We used this fact many times when discussing low-rank approximation. Show that $\mathbf{V}\mathbf{V}^T = (\mathbf{V}\mathbf{V}^T)(\mathbf{V}\mathbf{V}^T)$. Why does this property make intuitive sense if $\mathbf{V}\mathbf{V}^T$ is a projection?

(g) Show that if $v \in \mathbb{R}^n$ is an eigenvector of $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenvalue λ, then so is $c \cdot v$ for any scalar c.

(h) Show that for any symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1(\mathbf{A}), \ldots, \lambda_n(\mathbf{A})$, and integer p, the eigenvalues of $\mathbf{A}^p$ are equal to $\lambda_1(\mathbf{A})^p, \ldots, \lambda_n(\mathbf{A})^p$.

3. Let $\mathbf{X} \in \mathbb{R}^{d \times d}$ have eigendecomposition $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ with eigenvalues $\lambda_1(\mathbf{X}), \ldots, \lambda_d(\mathbf{X})$.

   (a) What are the eigenvalues of the matrix $\mathbf{X}^2 + \mathbf{X}^3$?

   (b) What are its eigenvectors?

4. What is one reason why you would want to compute a low-rank approximation of a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$?

5. $\mathbf{X} \in \mathbb{R}^{500 \times 50}$ contains 500 well-clustered data points as its rows. In particular, there are ten cluster centers $\vec{y}_1, \ldots, \vec{y}_{10} \in \mathbb{R}^{50}$, such that each row $\vec{x}_i$ lies within Euclidean distance at most 1 of a center. Give an *upper bound* on $\min_{\mathbf{B}:\text{rank}(\mathbf{B})=10} \|\mathbf{X} - \mathbf{B}\|_F^2$.

6. Let $\mathbf{X} \in \mathbb{R}^{n \times 900}$ have random entries drawn independently in $[0, 1]$. Let $\mathbf{Y} \in \mathbb{R}^{n \times 900}$ have rows corresponding to 30×30 pixel grayscale images of handwritten digits. Let $\mathbf{Z} \in \mathbb{R}^{n \times 900}$ have rows corresponding to 30×30 pixel grayscale images of handwritten letters from the English alphabet. All entries of $\mathbf{Y}$ and $\mathbf{Z}$ are in $[0, 1]$.

   (a) How do you expect $\sum_{i=11}^{900} \lambda_i(\mathbf{X}^T\mathbf{X})$, $\sum_{i=11}^{900} \lambda_i(\mathbf{Y}^T\mathbf{Y})$, and $\sum_{i=11}^{900} \lambda_i(\mathbf{Z}^T\mathbf{Z})$ to compare? Explain why in a few sentences.

   (b) Plot a guess at what the spectrums of these three matrices might look like. Do not worry about the scale of the y axis.