COMPSCI 514: Final Review

1 Format/Details

Held in the lecture room (Thompson Hall 104) on 12/19, 10:30am - 12:30pm. Will be aimed to have similar length to the midterm (1 hour, 15 minutes) but you can use the full 2 hours if needed.

Format/difficulty will be similar to the midterm, with a mix of short answers with explanations and problem solving. Likely will have four main questions and a fifth bonus question. The bonus will likely be on material covered in the last three classes.

No calculators, cheatsheets, or other aids are permitted.

2 Concepts to Study

Probability and Randomized Algorithms (First Half of Class)

- The exam will not specifically test this part of the class, but should be able to apply foundational techniques. E.g., compute expectations, linearity of expectation, union bound, etc.

Low-Rank Approximation and PCA

- Understand and apply important linear algebraic manipulations used. E.g.:
  - $y^T y = \|y\|_2^2$ and using this to split $\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^T y$.
  - $\text{tr}(\mathbf{A}\mathbf{A}^T) = \text{tr}(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}\|_F^2 = \sum_{i=1}^{\text{rank}(\mathbf{A})} \sigma_i(\mathbf{A})^2$.
  - 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 eigenvectors.
- 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.
- PCA: 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 – eigenvectors (principal components) as directions of greatest variance, measuring the quality of the optimal low-rank approximation by plotting the eigenvalues (the spectrum).

- Singular value decomposition definition.
- Connection of SVD of $\mathbf{X}$ to eigendecompositions of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$. Connection of singular values to eigenvalues of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$.
- Computing PCA/optimal low-rank approximation from the SVD. Connection of left and right singular vectors to the dual view of low-rank approximation as row and column approximation.
- Application of the SVD to linear regression (as seen on Problem Set 3).
- Low-rank approximation of a similarity matrix and entity embeddings (high level idea, don't need to know details).
- Iterative (sparse) vs. direct methods for SVD. Power method and high level ideas of analysis.

Spectral Methods for Graphs

- Adjacency matrix $\mathbf{A}$ and Laplacian ($\mathbf{L} = \mathbf{D} - \mathbf{A}$) definitions.
- Motivation behind using the second smallest eigenvector of the Laplacian to find a small but balanced cut. $\vec{x}^T\mathbf{L}\vec{x}$ as giving the size of a cut when $\vec{x} \in \{-1, 1\}^n$ is a cut indicator vector.
- Graph clustering for non-linearly separable data and for community detection.
- Stochastic block model definition, expected adjacency matrix, Laplacian, and eigenvectors. Why spectral clustering works for stochastic block model.
- Do not need to know matrix concentration proof, but should understand the high level idea. Why accuracy does down as q gets close to p.
- Do not need to know connection of power method to random walks, since we did not really get to cover it in detail.

Optimization

- How continuous optimization arises in machine learning through loss minimization.
- Definition of gradient and connection to directional derivative.
- Gradient descent.
- Convex function definition and corollary of what it implies about the gradient.
- Lipschitz function definition.
- Would not need to recreate the analysis of GD for convex Lipschitz functions and do not need to memorize the convergence theorem, but should understand the main ideas. Would be valuable to work through.
- Convex set definition, definition of projection, projected gradient descent for constrained optimization and why its analysis is essentially identical to that of gradient descent.

Online And Stochastic Gradient Descent

- Online optimization set up and online gradient descent.

- Regret definition. Why regret can be negative.
- Don't need to recreate OGD analysis or memorize the regret bound, but should understand the main ideas and how it compares to regular GD analysis.
- Stochastic gradient descent and why it can be analyzed as a special case of OGD.
- Again, don't need to recreate analysis or memorize convergence bound – but should understand main ideas. Should e.g., understand how $\theta^\star, \theta^{ol}$, and $\hat{\theta}$ differ/compare.
- Intuition behind when stochastic gradient descent takes many more iterations than gradient descent and when it doesn't.
- Gradient descent for least squares regression – how GD and SGD compare.

Other Topics

- Packing exponentially many random vectors in high-dimensional space.
- Why random points in high-dimensional space are likely to be very far from each other.
- Concentration of volume around the surface and equators of high-dimensional balls.
- How high-dimensional cubes and balls differ.
- Over-constrained verses under-constrained regression and compressed sensing set up.
- Sparse recovery definition and connection to Kruskal rank.
- Sparse recovery as a non-convex optimization probem. Convex relaxation and basis pursuit.
- Connection of sparse recover to the frequent items problem.

3 Practice Questions

Evolving. Check back to see if more questions have been added.

If you have time, I recommend trying to solve some problems first *without any resources or notes* first (like you would on an exam). Then, if you get stuck, go back to resources.

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 (how does $\mathbf{B}$ here connect to Problem 2.1 of Problem Set 3?), 3.22, 3.26, 7.16, 12.31, 12.33 *Foundations of Data Science.*
2. Linear algebra practice (some off Piazza):

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

   (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 rank rank$(\mathbf{A}) = k$, show that rank$(\mathbf{X}) \leq k$.

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

(d) Show that for any matrix $\mathbf{A}$ with SVD $\mathbf{U\Sigma V}^T$,

$$\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T\mathbf{A}) = \text{tr}(\mathbf{A}\mathbf{A}^T) = \|\mathbf{U\Sigma}\|_F^2 = \|\mathbf{V\Sigma}\|_F^2 = \sum_{i=1}^{n} \sigma_i(\mathbf{A})^2,$$

where $\sigma_i(\mathbf{A})^2$ is the i^{th} singular value of $\mathbf{A}$ (the i^{th} diagonal entry of $\mathbf{\Sigma}$) squared.

(e) Prove that if $\mathbf{V} \in \mathbb{R}^{d\times k}$ has orthonormal columns, then for any matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\|\mathbf{X} - \mathbf{XVV}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{XVV}^T\|_F^2$. Hint: Use that $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{AA}^T)$ for any $\mathbf{A}$ and that trace is linear: $\text{tr}(\mathbf{A}+\mathbf{B}) = \text{tr}(\mathbf{A}) + \text{tr}(\mathbf{B})$. The above is often called the 'pythagorean theorem'. I used it in class when deriving PCA. Why intuitively is it like the pythagorean theorem you are used to?

(f) For any $\mathbf{V} \in \mathbb{R}^{d\times k}$ with orthonormal columns, $\mathbf{VV}^T$ 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 $\mathbf{VV}^T = (\mathbf{VV}^T)(\mathbf{VV}^T)$. Why does this property make intuitive sense if $\mathbf{VV}^T$ is a projection?

(g) **More challenging:** Prove that for any $\mathbf{V} \in \mathbb{R}^{d\times k}$ with orthonormal columns $\mathbf{VV}^T$ is actually the projection onto the subspace spanned by the columns of $\mathbf{V}$ ($\mathbf{V}$'s column span). Hint: Formally to prove this, argue that for any vector $x \in \mathbb{R}^d$: $y = \mathbf{VV}^T x$ satisfies: $y = \arg\min_{z: z\in colspan(\mathbf{V})} \|x - z\|_2^2$.

3. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ have SVD $\mathbf{U\Sigma V}^T$ with singular values $\sigma_1(\mathbf{X}), \ldots \sigma_d(\mathbf{X})$. What are the eigenvalues of $(\mathbf{X}^T\mathbf{X})^q$? What are its eigenvectors? How about $(\mathbf{XX}^T)^q$. What is the runtime required to apply either of these two matrices to a vector?

4. Prove that for any matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and vector $y \in \mathbb{R}^d$, $\|\mathbf{X}y\|_2 \leq \sigma_1(\mathbf{X})$. **Hint:** Use Courant-Fischer: that the top eigenvector if a symmetric matrix $\mathbf{A}$ is given by $v_1 = \text{argmax}_{v:\|v\|=1} v^T\mathbf{A}v$.

5. Let $\mathbf{X} \in \mathbb{R}^{n\times 900}$ have random entries drawn independently as $\{0.1\}$. Let $\mathbf{Y} \in \mathbb{R}^{n\times 900}$ have rows corresponding to 30×30 pixel black and white images of handwritten digits. All entries of $\mathbf{Y}$ are in $\{0,1\}$. How do you expect $\sum_{i=11}^{30} \sigma_i(\mathbf{X})^2$ and $\sum_{i=11}^{30} \sigma_i(\mathbf{Y})^2$ to compare? Plot a guess what what the spectrums of these two matrices might look like (just looking for a high level idea, no real 'right answer').

6. You have a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ where each row corresponds to a student and each column corresponds to their grade on an assignment. The final column is their cumulative grade. What do you expect the rank of this matrix to be? Do you think it is well approximated by an even low-rank matrix?

7. Let $\mathbf{B} \in \mathbb{R}^{n\times d}$ be a rank-k matrix with SVD $\mathbf{B} = \mathbf{U\Sigma V}^T$. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be some other matrix. Which is smaller $\|\mathbf{X} - \mathbf{B}\|_F$ or $\|\mathbf{X} - \mathbf{UU}^T\mathbf{X}\|_F$? How about $\|\mathbf{X} - \mathbf{XVV}^T\|_F$?

8. Consider two matrice $\mathbf{A} = \begin{bmatrix} 1.01 & 0 \\ 0 & 1 \end{bmatrix}$ or $\mathbf{B} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1 \end{bmatrix}$? What are their eigenvalues and eigenvectors? On which matrix will power method converge more quickly?

Spectral Methods for Graphs

1. Argue that the minimum cut problem on a graph G with n nodes and Laplacian $\mathbf{L}$ is equivalent to solving $\min_{x\in\{-1,1\}^n} f(x)$ where $f(x) = x^T\mathbf{L}x$. Show that this objective function is convex. Is min-cut a convex optimization problem over a convex constraint set?

2. Consider the related problem: $\min_{x:\|x\|=1} f(x)$. Is this problem convex? What is the optimum? How does this relate to our use of the second smallest Laplacian eigenvector v_{n-1} in spectral clustering.

3. In the stochastic block model, why is clustering with the second largest eigenvector of the expected adjacency matrix equivalent to clustering with the second smallest eigenvector of the expected Laplacian? Are these two approaches identical when clustering using the actual rather than the expected matrices? Describe a natural variant of the stochastic block model where these two algorithms would not be equivalent even on the expected matrices.

Optimization/Gradient Descent

1. The sum of two convex functions $f(x)$ and $g(x)$ (i.e., $[f+g](x)$) is also convex. Always? Sometimes? Never?

2. The different of two convex functions $f(x)$ and $g(x)$ (i.e., $[f-g](x)$) is also convex. Always? Sometimes? Never?

3. The composition of two convex functions $f(x)$ and $g(x)$ (i.e., $[f \circ g](x)$) is also convex. Always? Sometimes? Never?

4. The union of two convex sets $A \cup B$ is also convex. Always? Sometimes? Never?

5. The sum of two G-Lipschitz functions is $2G$-Lipschitz. Always? Sometimes? Never?

6. The sum of two G-Lipschitz functions is G-Lipschitz. Always? Sometimes? Never?

7. Consider two vectors $x, y \in \mathbb{R}^d$ and let $\bar{x} = x \cdot \frac{\|y\|_2}{\|x\|_2}$. Show that $\|\bar{x} - y\|_2 \leq \|x - y\|_2$. Use an argument based around convex sets and projection.

8. Consider optimizing $f_1(x) = x^2$, $f_2(x) = (x-1)^2$ and $f_3(x) = (x+1)^2$ in an online fashion. What is θ^{ol}. What is the regret for the sequence of solutions $\theta^{(1)} = 0$, $\theta^{(2)} = .5$, $\theta^{(3)} = -.5$.

9. At the beginning of each day you provision servers for your website, enough to serve r requests. You pay \$.01 per request provisioned. During the day, $\hat{r}$ requests actually come in. If you do not have the capacity to serve a request, you must pay \$.03 to have the request served by an outside provider. At the end of the day, you can change the number of servers you have provisioned, to try to do better the next day. Formulate this problem over a 100 day time horizon as an online optimization problem. What will be the online gradient descent update each day?

10. High level/informal: Do you expect stochastic gradient descent to perform better or worse in comparison to gradient descent in solving linear regression when the data matrix $\mathbf{X}$ is close to low-rank or not close to low-rank?

High-Dimensional Geometry and Sparse Recovery

1. High-dimensional geometry: Exercises 2.8, 2.9, 2.10, 2.13, 2.20, 2.24, 2.25, 2.26, 2.28, 2.34. *Foundations of Data Science.*