CS168, Spring 2017

CS168 Final Exam

(Do not turn this page until you are instructed to do so!)

Instructions: This is a closed-book exam, and you are permitted to refer only to one double-sided sheet of notes, which you should have prepared in advance (though the notes should not be necessary to complete the exam). You have 3 hours and the exam is worth 168 points. Make sure you print your name legibly and sign the honor code below. All of the intended answers can be written well within the space provided. You can use the back of the preceding page for scratch work. If you want to use the back side of a page to write part of your answer, be sure to mark your answer clearly. Good luck!

The following is a statement of the Stanford University Honor Code:

A. The Honor Code is an undertaking of the students, individually and collectively:

(1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

(2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

B. The faculty on its part manifests its codence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

C. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

_____

(Signature)

_____

(Print your name, legibly!)

Problem	#1	#2	#3	#4	#5	#6	#7	#8	#9	Total
Score										
Maximum	30	15	20	15	17	18	13	12	28	168

1. **Miscellaneous short answer.** (30 points)

(a) (5 points) Which of the following functions are convex? You do not need to provide a proof, just circle the functions that are convex:

i. x^3

ii. $-3\sqrt{x}$ for $x \geq 0$

iii. the ℓ_0 norm of a vector in $\mathbb{R}^n$

iv. the ℓ_1 norm of a vector in $\mathbb{R}^n$

v. the ℓ_∞ norm of a vector in $\mathbb{R}^n$

(b) (5 points) Which of the following sets are guaranteed to be convex? You do not need to provide a proof, just circle the convex sets:

i. the set $\{(x, y) \in \mathbb{R}^2 : 1 \leq 2x^2 + 3y^2 \leq 4\}$

ii. the vectors of $\mathbb{R}^n$ that have ℓ_2 norm at least 1

iii. the set of $n \times n$ symmetric matrices, viewed as a subset of $\mathbb{R}^{n^2}$

iv. an arbitrary intersection of convex sets

v. an arbitrary union of convex sets

(c) (5 points) What are the eigenvalues of the Laplacian matrix of the complete graph on n vertices? You do not need to prove your answer.

(d) (5 points) Suppose G is a d-regular graph (i.e., every vertex has d incident edges). What is the relationship between the eigenvalues of the adjacency matrix of G, and the eigenvalues of the Laplacian matrix of G? You do not need to prove your answer.

(e) (5 points) What is the largest-possible fraction of the population that can have income at least five times the average income of the population? (Assume that income cannot be negative, and that the average income is not 0.) Describe a hypothetical population/distribution over incomes where this fraction is achieved.

(f) (5 points) Write a linear program that expresses the following optimization problem over variables x and y: Minimize $|x + 1| + |x + y - 2|$ subject to the constraints that $2x + y \leq 6$ and $y - x \geq -3$. [Hint: the linear program can involve more variables than just x and y.]

2. **Hashing.** (15 points) In Lecture 2, we studied the count-min sketch, which allows us to approximate the number of times an element occurs in a data stream, using much less space than it would take to store the count for every observed element. Recall that the count-min-sketch data structure is defined in terms of t hash functions $h_1, \ldots, h_t$ that each map an element x to one of b buckets, and hence the count-min-sketch data structure can be thought of as a $t \times b$ matrix of numbers, $CMS[i][j]$ where $i \in \{1, \ldots, t\}$ and $j \in \{1, \ldots, b\}$. The datastructure supports two operations: `Inc(x)` and `Count(x)`. `Inc(x)` is called whenever we see element x in the data stream:

```
def Inc(x):
    for i = 1, 2, ..., t:
        increment CMS[i][h_i(x)]
```

`Count(x)` returns the estimated number of times that we've seen element x: $\min_{i=1}^{t} \text{CMS}[i][h_i(x)]$.

(a) (3 points) Why is it better for `Count(x)` to return the minimum instead of the average of the t values?

(b) (6 points) If $h_i(x) = h_i(y)$ for some values x and y and some hash function h_i, then x and y are said to *collide* under h_i. Collisions are the cause of estimation errors; what goes wrong if we try to pick hash functions h_i such that there are *no* collisions between any two values in the input data stream?

(c) (6 points) We have t hash functions and b buckets for each of those hash functions. Suppose we define the hash functions as follows:

$$h_i(x) = (x + i) \mod b.$$

Is this a good choice for the set of hash functions? Why or why not?

3. **Similarity search.** (20 points) You decide to build an image search system using locality sensitive hashing (recall Mini-Project #2 where you made a locality sensitive hashing based search system for the newsgroups documents). Each image is represented as a vector, scaled so that it has unit length. To measure similarity, you decide to use the angle metric, defined by $\arccos(x \cdot y)$ for vectors x and y and denoted by $angle(x, y)$—this is simply the angle between the vectors x and y. Throughout this problem, $x \cdot y = \langle x, y \rangle$ denotes the inner product of vectors x and y. For some specified angle θ, the goal is: given a query x, if there is an image y in the system with $angle(x, y) \leq \theta$, then the system should return an image z with $angle(x, z) \leq 2\theta$.

   (a) (4 points) Consider the hash function $h(x) = sign(g \cdot x)$, where g is a vector with i.i.d standard Gaussian entries and $sign(c)$ is either ± 1 according to whether $c \geq 0$. If $angle(x, y) \leq \theta$, then what can you say about the probability (over the choice of g) that $h(x) = h(y)$? [Hint: what is the probability that $h(x) = h(y)$ if $angle(x, y) = \theta$?]

   (b) (4 points) If $angle(x, y) \geq 2\theta$, what can you say about the probability that $h(x) \neq h(y)$?

   (c) (4 points) Define $H(x) = (h_1(x), h_2(x), \ldots, h_d(x))$ by concatenating d independent hash functions $h(x)$ chosen as specified above. If $angle(x, y) \leq \theta$, what's the probability that $H(x) = H(y)$? If $angle(x, y) \geq 2\theta$, what's the probability that $H(x) \neq H(y)$?

   (d) (4 points) Whatever our choice of d, suppose we use $\ell = (1 - \theta/\pi)^{-d}$ independent hash tables. Prove that if $angle(x, y) \leq \theta$, then the probability that $H(x) = H(y)$ for at least one hash table is at least $1 - \frac{1}{e}$. [Hint: feel free to assume, without proof, that for any $c \in (0, 1), (1 - c)^{1/c} < \frac{1}{e}$.]

   (e) (4 points) Now set $d = \frac{\log n}{-\log(1-2\theta/\pi)}$ and set ℓ as above. Prove that the number of irrelevant images—images y with $angle(x, y) \geq 2\theta$—such that $H(x) = H(y)$ for at least one hash table is at most 4ℓ with probability at least $3/4$, and hence this approach to finding similar images will not spend too much time sorting through dissimilar images.

4. **Generalization and regularization.**(15 points) You are working with a team of computation biologists who are trying to build model to predict a phenotype of an individual, such as their risk of diabetes, from their gene sequence. A dataset of the gene sequence of m individuals has been complied for this purpose. For each individual in the dataset, you have their DNA sequenced at $d = 10^6$ locations. The biologists in your team feel that mutations at certain *pairs* of DNA locations should together be responsible for the phenotype, hence you want to experiment with developing a linear classifier that has a coefficient for each pair of DNA locations.

   (a) (5 points) Without using regularization, roughly how much training data would you need to fit such a linear model that generalizes well beyond the training data?

   (b) (5 points) How would your answer change if you knew that that ground truth prediction function depends on only a limited number k of gene pairs?

   (c) (5 points) Suppose you don't have the budget to sequence so many individuals. How can you still train your model while ensuring that it does not overfit? What exactly would you do to encourage a sparse solution?

5. **PCA vs. JL.** (17 points)

(a) (5 points) Briefly explain how the PCA and Johnson-Lindenstrauss (JL) approaches to dimensionality reduction work.

For each of the following parts, specify whether PCA or JL would be the more appropriate dimensionality reduction method, and give a 1–2 sentence explanation for your answer.

(b) (3 points) You want to visualize your data.

(c) (3 points) You suspect that your data has low-dimensional linear structure.

(d) (3 points) The method of data collection that you were using introduced a large amount of noise along some specific direction or low-dimensional subspace.

(e) (3 points) You want to use a nearest neighbor subroutine to explore your data and classify new data.

6. **Singular value decomposition.** (18 points)

(a) (4 points) Consider the following grayscale image of the Moon. Draw a sketch of your best guess for what the best rank-1 approximation of the image would look like in the space indicated below. Assume that the representation of the image stores *black as zero*.

Figure 1: The Moon

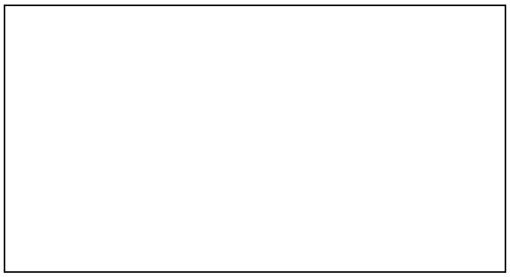

Figure 2: Sketch the best rank-1 approx. here!

(b) (3 points) *Netflix* has an enormous dataset of ratings of movies given by its users, and it needs your help to analyze it. Assume there are 10000 users and 1000 movies, and each user has rated every movie. This data can be represented as a 10000×1000 dimensional matrix M, where each entry $M(i, j)$ denotes the rating given by user i to movie j.

Suppose you perform an SVD $M = UDV^T$ of the matrix M. What concepts might be captured by the top few right singular vectors (the top singular vectors are the ones corresponding to the largest singular values)? What concepts might be captured by the top few left singular vectors?

(c) (3 points) Continuing the previous part, you carry out a SVD of M, and observe that the top right singular vectors do seem to correspond to interesting concepts. You want to perform a low-dimensional projection of the 10,000 users, to visualize them in the 2-dimensional space spanned by the top 2 right singular vectors. How can you use the SVD to directly find the projection of each user (i.e. each row of the matrix M) onto the top 2 right singular vectors, without actually computing the inner product?

(d) (4 points) Assume now that we are in the more realistic setting where all the $10000 * 1000$ ratings corresponding to every (user,movie) pair are not available. Assume that 10% of the entries are missing. Low rank approximation can be used in this case to complete the matrix M to infer the missing entries. You use the following algorithm to infer the missing entries: 1) You first set them to be the average overall rating to obtain a new matrix $\hat{M}$. 2) You find a rank-k approximation $\tilde{M}$ of the matrix $\hat{M}$. 3) You set the missing entries to be their value in $\tilde{M}$. Explain: i) How and ii) Why can you use the top k singular vectors to find a rank-k approximation.

(e) (4 points) Using the approach outlined in the previous part, how do you expect the error in estimating the missing entries to vary as a function of k? Explain.

7. **Markov Chains.** (13 points) Alice's childhood dream is to become a weather forecaster. Over several years, she collects data on whether it rains each day or not. She observes that if it rains one day, then there is a 60% probability that it rains the following day. On the other hand, if it does not rain one day, the probability that it rains the next day is only 10%.

   (a) (2 points) What is the size of the transition matrix corresponding to Alice's model?

   (b) (4 points) What is the largest eigenvalue of the transition matrix corresponding to Alice's model?

   (c) (3 points) Given that it rains today, what is the probability that it also rains three days from now, according to Alice's model?

   (d) (4 points) What is the asymptotic running time for calculating the probability that it rains n days from now, given that it rains today? Explain how you can achieve this running time.

8. **Convolutions.** (12 points) Suppose you are recording the mating calls of a species of rare whale. You suspect that your recording device is faulty, and realize that if the true signal is s, the device ends up recording the convolution $s * f$, for some filter f.

   (a) (3 points) Suppose you somehow figure out the filter f. You have the recording $r = s * f$, and wish to obtain the true signal s. How do you compute this?

   (b) (3 points) Assuming some measurement imprecision (e.g. you are only recording r to 16 bits of precision), what properties of f determine the extent to which you will be capable of recovering an accurate estimate of s from $r = s * f$? (Namely, when might you not be able to accurately recover s from r?)

   (c) (3 points) You now wish to figure out what the filter f actually is. Given any signal t, your recording device returns $t * f$. What signal t could you play into the recording device, such that you recover the filter f? (i.e. what vector t has the property that $t * f = f$?)

   (d) (3 points) You do the above experiment, and realize that the filter f is just a series of delta function—namely the recording device just adds several "echoes" of the true signal s. In general should you expect to be able to invert this convolution and obtain an accurate representation of the mating call, s, from the recording $s * f$? Why or why not?

9. **The Algorithmic Toolbox.** (28 points) For each of the following scenarios, select the technique/tool from the list that is best suited to the problem. Note that some tools/techniques might be matched with several scenarios, and some tools might not be matched with any scenarios. For each question, identify a tool/technique, and **provide a one-sentence explanation of how or why the technique should be applied.**

   Reminder of tools/techniques: Consistent Hashing, Count-Min-Sketch, Dimension Reduction, k-d Trees and Nearest Neighbor Search, regularization (ℓ_1 and ℓ_2), PCA, SVD and low-rank matrix approximation, tensor methods and low-rank tensor decomposition, Spectral Graph Theory, Importance Sampling, Markov Chain Monte Carlo, Fourier Analysis/Convolution, Compressive Sensing, Linear/Convex Programming.

   (a) (4 points) As the lead data scientist investigating the dispersion pattern of pollution generated at a set of several thousand power plants scattered across the country, you want to predict the amount of pollution that will be present at each of several thousand cities, and also estimate the amount of that pollution that was produced $t = 0, 1, 2, \ldots$ days ago. As a first pass, you assume that the spread of pollution can be modeled via a translation-invariant diffusion process. How could you make the computation of this pollution dispersion extremely efficient (perhaps with the goal of being able to answer questions like "what happens if we move this power station here?" in real-time).

   (b) (4 points) Impressed by your efficient pollution model, you are given extra funding to make a hyper-accurate (but possibly more computationally intensive) model for several metropolitan areas, and want to take into account the local geography, predominant wind patterns, etc. What techniques could you use to do this?

(c) (4 points) Your medical-devices startup has been focussing on low-power microprocessing for medical applications. One product that you are working on is a new hearing-aid that performs a large amount of signal processing and machine learning (e.g. learning the voices of people you talk with frequently, and then selectively filters/amplifies their voices, etc.), in addition to amplifying the volume. Because these devices need to be low-power (so they don't heat up), you want to do as much computation as possible on compressed representations of the incoming audio signals. What tools might be helpful?

(d) (4 points) Partway through your astronomy PhD, you realize that most of the cost and energy expenditure of the large telescope you are designing has to do not with the actual photon detector array, but with the cost of storing the incoming information in a high-throughput fashion. You decide to make an integrated circuit that takes, as input, the stream of x, y coordinates of incoming photos, and wish to output a final image that captures the locations that have the highest number of incoming photons, and maybe ignores many of the pixels that have very low/background levels of incoming photons. What tool might you use to approach this novel telescope design problem?

(e) (4 points) After four years of undergrad cooped up in libraries doing psets, you decide to reconnect with the outdoors and the earth, and join an organic farming co-op in northern Vermont. At the co-op, you are trying to develop a new hybrid type of tomato that is delicious and requires little water. The co-op has collected seeds from 100 different heirloom tomato varieties, and you would like to predict how tasty and how water-resistant any given hybrid of a pair of these types would be. The other folks at the co-op suggest trying to raise all 100^2 possible pairings. You have a better idea for being able to predict the quality of any such pairing. 1) What type of structure might you expect to be present in the 100×100 matrix depicting the quality of each hybrid? 2) How might you leverage that structure to avoid trying out all 100^2 hybrids?

(f) (4 points) You are hired as the first head of HR of a rapidly growing startup that now has 80 employees, and is about to move into a new office space. Your first task is to make the seating-plan for the new office, and you would like to assign desks in such a way that people who work closely together will tend to be sitting near each other. Because this is a startup, this task is complicated by the fact that many employees have multiple roles within the company (e.g. Tegan is both the data-structures guru, as well as sometimes helps Mike the graphics person with the front-end user interface; smooth talking Sally both does the rapid prototyping, as well as works with the sales team to pitch industry clients, etc.). To begin, you have all the employees send you a list of the colleagues they interact with the most. What could you do next to help you devise a great seating plan?

(g) (4 points) You found a small hedge fund with your cs168 project partners. To start, you train a deep neural network to accurately evaluate the quality (e.g. ratio of return to the risk) of a given stock portfolio. Specifically, given a hypothetical portfolio of stocks, represented as a vector v whose ith coordinate corresponds to the amount of the ith stock in the portfolio, the deep network will output a number $f(v)$ corresponding to the quality of the portfolio. Given this tool that you have developed, how might you find a near-optimal portfolio v^{opt} to invest in?

[this page intentionally left blank]