Princeton University

# ORF 363/COS 323
# Final Exam, Fall 2023

December 16, 2023

*Instructor:* A.A. Ahmadi

*AIs:* Beneventano, Chaudhry, Hua, Li, Lok

1. Please write your name on the first page of your solutions. Next to it, please write out and sign the following pledge: "I pledge my honor that I have not violated the Honor Code or the rules specified by the instructor during this examination."

2. The exam is not to be discussed with *anyone* except possibly the instructors and the AIs. You can only ask *clarification questions*, and only as *public* (and preferably non-anonymous) questions on Ed Discussion. No emails.

3. You are allowed to consult the lecture notes, your own notes, the reference books of the course as indicated on the syllabus, the problem sets and their solutions (yours and ours), the midterm and its solutions (yours and ours), the practice midterm and final exams and their solutions, all Ed Discussion posts, but *nothing else*. You can only use the Internet in case you run into problems related to software.

4. You may refer to facts proven in the notes or problem sets without reproving them.

5. For computational problems, include your code. The output you present should come from your code. Report requested numerical values to 4 digits after the decimal point.

6. You have 48 hours from the time of download to submit this exam on Gradescope as a *single PDF file*. The latest submission time is Thursday (December 21, 2023) at 10PM EST. You are free to write your solutions on paper or on a tablet, or to type them up. Only the latest version submitted before your deadline will be graded.

7. Each question has 25 points. You need to justify your answers to receive full credit.

**Problem 1: Setting the odds in your favor with semidefinite programming**

As the CEO of TigerCasino in Vegas, you are introducing a new game on your floor. In this game, a player rolls a die twice and receives $q_{ij}$ dollars from the casino if the die shows $i$ on one roll and $j$ on the other (the order does not matter). The matrix $Q = (q_{ij})_{1 \le i,j \le 6}$ is announced to the player:

$$Q = 100 \times \begin{pmatrix} 4 & -2 & 1 & -1 & -2 & 1 \\ -2 & 4 & 1 & -2 & -2 & -1 \\ 1 & 1 & 4 & -2 & 1 & -1 \\ -1 & -2 & -2 & 4 & 1 & -1 \\ -2 & -2 & 1 & 1 & 4 & -1 \\ 1 & -1 & -1 & -1 & -1 & 4 \end{pmatrix}.$$

Gamblers are leaving the Bellagios and rushing to your table because if the die was fair (i.e., had a probability of $\frac{1}{6}$ assigned to each outcome of a roll), they would make \$11.11

in expectation in every play. Little do they know, however, that you have been using your optimization knowledge to optimally bias the die and maximize the profit of TigerCasino. Let $x_i$ be the probability that the die comes out $i$. The optimization problem of interest to you is:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^6} \quad & x^T Q x \\
\text{s.t.} \quad & x \geq 0 \\
& \sum_{i=1}^{6} x_i = 1.
\end{aligned}
\tag{1}
$$

The constraints make sure that $x$ is a valid probability vector and the objective function is the expected payoff of the player in every play.

(a) Is problem (1) a convex optimization problem? Why or why not?

(b) Recall that a matrix $A \in \mathbb{S}^{n\times n}$ is copositive if $x^T A x \geq 0$ for all $x \geq 0$. Denote the set of $n \times n$ copositive matrices by $\mathcal{C}_n$. Show that the optimal value of (1) is equal to the optimal value of the following problem:

$$
\begin{aligned}
\max_{t \in \mathbb{R}} \quad & t \\
\text{s.t.} \quad & Q - tJ \in \mathcal{C}_6.
\end{aligned}
\tag{2}
$$

Here, $J \in \mathbb{S}^{6\times 6}$ is the all-ones matrix.

(c) Denote the optimal value of (1) (or equivalently (2)) by $OPT$. Denote the optimal value of the semidefinite program

$$
\begin{aligned}
\max_{t \in \mathbb{R}, N \in \mathbb{S}^{6\times 6}} \quad & t \\
\text{s.t.} \quad & Q - tJ - N \succeq 0 \\
& N \geq 0
\end{aligned}
\tag{3}
$$

by $SDP_{OPT}$. (Here, "$\geq$" denotes an entrywise nonnegativity constraint and "$\succeq$" denotes a positive semidefiniteness constraint.) Show that $SDP_{OPT} \leq OPT$.

(d) Report $SDP_{OPT}$ by solving (3) in `cvx` or `cvxpy`. Show that $SDP_{OPT} = OPT$ by presenting a vector $x^* \in \mathbb{R}^6$ that is feasible to (1) and makes the objective function of (1) equal to $SDP_{OPT}$. (Hint: you may wish to start with an eigenvector associated with the smallest eigenvalue of $Q - t^*J - N^*$, where $(t^*, N^*)$ form an optimal solution to (3).) What probability does your optimal die assign to each of its six outcomes? What is the expected win/loss of TigerCasino in dollars every time a player plays this game?


**Problem 2: Would your GPA be higher at Yale?**

An article appeared earlier this month in the New York Times with the title "*Nearly Everyone Gets A's at Yale. Does That Cheapen the Grade?*"[1] After reading the article, you may wonder whether it is easier to get an A at Yale than at Princeton and, if so, how one could adjust GPAs to account for course difficulty. In this problem, we approach this question using an optimization-based idea proposed by Professor Vanderbei[2] and his collaborators.

[1] `https://www.nytimes.com/2023/12/05/nyregion/yale-grade-inflation.html`

[2] We also take this opportunity to honor Professor Vanderbei who is retiring this January after teaching at Princeton for 33 years.

We begin by asking four familiar Princeton/Yale students to take some courses at their own institution and a few similar ones at the other institution. The letter grades of these students are summarized in Table 1. Their GPAs are calculated using the grade points in Table 2.

| | Princeton ORF 363 | Princeton ENG 351 | Princeton ORF 309 | Yale CPSC 365 | Yale ENGL 305 | Yale STAT 241 | GPA | Aptitude |
|---|---|---|---|---|---|---|---|---|
| M. Obama | A- | A | B+ | | A+ | | 3.825 | ? |
| J. Bezos | A+ | A- | B- | A | | | 3.675 | ? |
| M. Streep | B- | | | A | A | A | 3.675 | ? |
| R. DeSantis | | | B- | A+ | A+ | A+ | 3.9 | ? |
| Inflatedness | ? | ? | ? | ? | ? | ? | | |

Table 1: Performance of four students in six courses

| Letter Grade | A+ | A | A- | B+ | B | B- |
|---|---|---|---|---|---|---|
| Grade Point | 4.3 | 4.0 | 3.7 | 3.3 | 3.0 | 2.7 |

Table 2: Converting letter grades to grade points

We assume that the grade point $g_{ij}$ that student $i$ receives in course $j$ should nearly be equal to $a_i + b_j$, where $a_i$ is the "*aptitude*" of student $i$ and $b_j$ is the "*inflatedness*" of course $j$. In our example, $i \in \{1, \ldots, 4\}$, and $j \in \{1, \ldots, 6\}$. We normalize the inflatedness scores with the constraint $\sum_{j=1}^{6} b_j = 0$ (negative inflatedness scores correspond to more difficult courses). This leads us to the following optimization problem which simultaneously computes student aptitudes and course inflatedness scores:

$$\begin{aligned} \min_{a \in \mathbb{R}^4, b \in \mathbb{R}^6} \quad & \textstyle\sum_{(i,j) \in \mathcal{G}} (g_{ij} - a_i - b_j)^2 \\ \text{s.t.} \quad & \sum_{j=1}^{6} b_j = 0. \end{aligned} \tag{4}$$

Here, the index set $\mathcal{G}$ denotes the student-course pairs for which a grade is available.

(a) Is problem (4) a convex optimization problem? Why or why not?

(b) Use Tables 1 and 2 to solve problem (4) via `cvx` or `cvxpy`. Fill in the question marks in Table 1 with your optimal solution.

(c) How do the four students rank based on their aptitude (which can be thought of as an "adjusted GPA")? Compare this to the GPA-based ranking. Which courses have the lowest/highest inflatedness score?

**Problem 3: Getting More Than You Receive**

Your friend would like to send a signal $x^* \in \mathbb{R}^n$ of length $n = 256$ to you by transmitting a much smaller vector $b \in \mathbb{R}^m$ with $m < n$ to save data. Fortunately, $x^*$ is sparse in some basis: you know that if $D \in \mathbb{R}^{n \times n}$ is the discrete cosine transform matrix (the details of which will not be important for this problem), then the vector $z = Dx^* \in \mathbb{R}^n$ has few nonzero entries. To compress $x^*$, your friend takes $m = 32$ measurements using a random matrix $A \in \mathbb{R}^{m \times n}$ (generated from a particular seed so that you have access to the same matrix), and transmits the vector $Ax^*$. On the other end of the channel, you receive the signal $b = Ax^* + e \in \mathbb{R}^m$, where $e \in \mathbb{R}^m$ represents some random noise introduced during transmission. Will you be able to recover $x^*$ from $b$; i.e., not only filter out the noise but also get much more than you receive? Let's see!

The following Python code generates the data $x^*$, $A$, $D$, and $b$ described above:

```
import numpy as np; from scipy.fftpack import dct
freq = np.array([29, 30, 196, 223])
x_star = np.array([sum(np.cos(2*np.pi*freq/256*(i+0.5))) for i in range(256)])
D = dct(np.eye(256), axis=0, type=2, norm="ortho")
np.random.seed(363)
A = (2*np.random.rand(32, 256) - 1) / np.sqrt(32 / 3)
b = A @ x_star + 0.05*(2*np.random.rand(32) - 1)
```

The following MATLAB code generates the data $x^*$, $A$, $D$, and $b$ described above:

```
freq = [29, 30, 196, 223];
x_star = zeros(256, 1);
for i = 1:256
    x_star(i) = sum(cos(2 * pi * freq / 256 * (i - 0.5)));
end
D = dct(eye(256));
rng(363, "twister");
A = (2*transpose(rand(256, 32)) - 1) / sqrt(32 / 3);
b = A * x_star + 0.05*(2*rand(32, 1) - 1);
```

(a) You first attempt to recover your friend's signal $x^*$ by finding a vector whose measurements are closest in $\ell_2$ norm to the signal $b$ that you received:

$$\min_{x \in \mathbb{R}^n} \quad \|Ax - b\|_2. \tag{5}$$

Does problem (5) have a unique solution? Prove that any solution to (5) solves the equations $A^T(b - Ax) = 0$, and hence conclude that the true signal $x^*$ is not optimal for this problem. Solve (5) using `cvx` or `cvxpy`, and compare the solution that you obtain against $x^*$ by plotting both vectors on the same figure.

(b) You remember that you can also minimize the $\ell_1$ norm[3] of $z = Dx$ to promote sparsity. Since you know that $A^T(b - Ax^*) \neq 0$, you relax this condition by constraining the $\ell_\infty$ norm of $A^T(b - Ax)$. This leads you to the following problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|Dx\|_1 \\ \text{s.t.} \quad & \|A^T(b - Ax)\|_\infty \leq 0.2. \end{aligned} \tag{6}$$

Show that problem (6) can be formulated as a linear program. You do not need to convert your linear program to standard form.

Solve problem (6) using `cvx` or `cvxpy`, and compare the solution that you obtain against $x^*$ by plotting both vectors on the same figure.

[3] Recall that the $\ell_1$ and $\ell_\infty$ norms of $x \in \mathbb{R}^n$ are defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$ and $\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|$.

**Problem 4: Getting to the bottom of a ski-friendly function**
A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *ski-friendly* if it satisfies

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu \cdot \big(f(x) - f^*\big) \tag{7}$$

for all $x \in \mathbb{R}^n$ and for some $\mu > 0$. Here, $\|\nabla f(x)\|$ denotes the two-norm of the gradient of $f$ at $x$ and $f^* \in \mathbb{R}$ denotes the minimum value of $f$. Ski-friendly functions appear in the study of certain families of neural nets. While not necessarily convex, they have certain attractive properties which we examine in this problem.

(a) An example of a univariate ski-friendly function is $g(x) = x^2 + 3\sin^2(x)$ (you are not required to verify this, but can plot the function for fun to see if you can ski on it). Show that this function is not convex.

(b) Show that every local minimum of a ski-friendly function is a global minimum. (This is the same property that convex functions enjoy.)

(c) Suppose in addition to being ski-friendly, a function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2$$

for all $x, y \in \mathbb{R}^n$ and for some scalar $L > 0$. Suppose we start at an initial point $x_0 \in \mathbb{R}^n$ and run gradient descent on $f$ with a constant step size of $\frac{1}{L}$:

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k).$$

Show that

$$f(x_1) - f^* \leq \beta \cdot \big(f(x_0) - f^*\big)$$

for a constant $\beta$ that only depends on $\mu$ and $L$. Give an expression for $\beta$ in terms of $\mu$ and $L$.

(d) Suppose $\mu$ and $L$ are such that $\beta \in (0, 1)$. How many iterations of gradient descent does it take to halve the optimality gap? In other words, write down the smallest integer $t$ (in terms of $\mu$ and $L$) that ensures

$$f(x_t) - f^* \leq \frac{1}{2}\big(f(x_0) - f^*\big).$$