MA5232 Assignment Part 4

Convex Programming

Alon Rafael Landmann

A0276620E

May 17, 2024

1 Geometric Programming

Part 1 (i)

A twice differentiable function is convex on $\mathbb{R}^n$ if and only if its Hessian matrix is positive semi-definite everywhere. We need to show this for $c > 0$ and

$$f(y_1, \dots, y_n) = c \exp(a_1 y_1 + \dots + a_n y_n).$$

Let $y \in \mathbb{R}^n$. Computing the necessary partial derivatives yields

$$\begin{aligned}
\partial_i f(y_1, \dots, y_n) &= c a_i \exp(a_1 y_1 + \dots + a_n y_n), \\
\partial_{ji} f(y_1, \dots, y_n) &= c a_i a_j \exp(a_1 y_1 + \dots + a_n y_n).
\end{aligned}$$

Let x be another vector in $\mathbb{R}^n$. Then

$$\begin{aligned}
\langle x, (H|_y) x \rangle &= \sum_{i=1}^n \sum_{j=1}^n (H|_y)_{ij} x_i x_j \\
&= \sum_{i=1}^n \sum_{j=1}^n c a_i a_j \exp(a_1 y_1 + \dots + a_n y_n) x_i x_j \\
&= c \exp(a_1 y_1 + \dots + a_n y_n) \sum_{i=1}^n \sum_{j=1}^n (a_i x_i)(a_j x_j) \\
&= c \exp(a_1 y_1 + \dots + a_n y_n) \left(\sum_{i=1}^n (a_i x_i) \right)^2
\end{aligned}$$

Each of the factors in the final expression is non-negative. Therefore, the Hessian matrix is positive semi-definite everywhere, completing the proof that f is convex.

Part 1 (ii)

Consider a general geometric program.

$$\begin{aligned}
\min \quad & \sum_{k=1}^N c_{k0} x_1^{a_{1k0}} \cdots x_n^{a_{nk0}} & \\
\text{s.t.} \quad & \sum_{k=1}^N c_{ki} x_1^{a_{1ki}} \cdots x_n^{a_{nki}} \leq 1, & \forall i = 1, \dots, m, \\
& c_j x_1^{a_{1j}} \cdots x_n^{a_{nj}} = 1, & \forall j = 1, \dots, p, \\
& x_1, \dots, x_n > 0, &
\end{aligned}$$

where all constants labeled c are positive real numbers and all constants labeled a are unconstrained real numbers.

Under the bijective change of variables $y_t = \log x_t$ we obtain the following equivalent formulation.

$$\begin{aligned}
\min \quad & \sum_{k=1}^N c_{k0} \exp(a_{1k0} y_1 + \dots + a_{nk0} y_n) & \\
\text{s.t.} \quad & \sum_{k=1}^N c_{ki} \exp(a_{1ki} y_1 + \dots + a_{nki} y_n) \leq 1, & \forall i = 1, \dots, m, \\
& c_j \exp(a_{1j} y_1 + \dots + a_{nj} y_n) = 1, & \forall j = 1, \dots, p, \\
& y_1, \dots, y_n \in \mathbb{R}. &
\end{aligned}$$

A convex program, is one in which we minimize a convex function over a convex constraint set. Let us first consider the objective function. By Part 1 (a) we need only show that the sum of two convex functions is again convex. By induction we then have that any finite sum of convex functions is also convex.

Lemma 1. *Let f and g be convex functions on $\mathbb{R}^n$. Then $f+g$, defined point-wise, is convex.*

Proof. Let $\theta \in (0,1)$, and $x, y \in \mathbb{R}^n$. Then

$$\begin{aligned}(f+g)(\theta x + (1-\theta)y) &= f(\theta x + (1-\theta)y) + g(\theta x + (1-\theta)y) \\ &\leq \theta f(x) + (1-\theta)f(y) + \theta g(x) + (1-\theta)g(y) \\ &= \theta(f+g)(x) + (1-\theta)(f+g)(y).\end{aligned}$$

□

To complete the definition of a convex program, we have to show that the constraint is a convex set. We do this by showing that it is an intersection of convex sets, which is again convex by the following argument.

Lemma 2. *Let C and D be convex subsets of $\mathbb{R}^n$. Then $C \cap D$ is a convex set.*

Proof. Let $\theta \in (0,1)$, and $x, y \in C \cap D$. Then clearly $x, y \in C$, and $x, y \in D$. And so by the convexity of C and D we have $\theta x + (1-\theta)y \in C$, and $\theta x + (1-\theta)y \in C$, respectively. Thus, $\theta x + (1-\theta)y \in C \cap D$. □

By a simple induction argument, we can again show that the same holds for finite intersections of convex sets. Therefore, if we can now show that the constraints separately form convex sets, we are done.

The inequalities are all of the form $g(x) \leq 1$ for some convex function $g : \mathbb{R}^n \to \mathbb{R}$ (by Part 1 (a) and Lemma 1). Let $x, y \in \mathbb{R}^n$ be two points that satisfy the inequality, and let $\theta \in (0,1)$. Then we have

$$g(\theta x + (1-\theta)y) \leq \theta g(x) + (1-\theta)g(y) \leq \theta \cdot 1 + (1-\theta) \cdot 1 = 1$$

showing that the sets defined by the inequalities are convex.

To conclude, we need to consider the equality constraints. Since the logarithmic function is bijective we can equivalently consider

$$\log c_j + (a_{1j}y_1 + \cdots + a_{nj}y_n) = 0$$

for every $j = 1, \ldots, p$. Considering all the equalities together and moving the logarithmic terms to the right-hand-side we obtain an affine constraint of the form

$$Ay = b,$$

where b is a constant vector. Affine constraint sets are known to be convex, as outlined in the examples in the lecture notes. We have thus shown that the geometric program has an equivalent convex formulation.

Part 1 (iii)

Taking x_1 to denote the height, x_2 the width, and x_3 the length of a box, we can formulate the box optimization problem as follows.

$$\begin{aligned}\max \quad & x_1x_2x_3 \\ \text{s.t.} \quad & 2x_1x_2 + 2x_1x_3 + 2x_2x_3 \leq a_1, \\ & x_2x_3 \leq a_2, \\ & r_1 \leq x_1/x_2 \leq r_2, \\ & r_3 \leq x_2/x_3 \leq r_4, \\ & x_1, x_2, x_3 > 0.\end{aligned}$$

Note that we have $a_1, a_2, r_2, r_4 > 0$ for the problem to make physical sense, and so we can divide by these constants instead of bringing terms like $(a_1 - 1)$ and $(a_2 - 1)$ to the left-hand-side, which would be more complicated. Carrying out the divisions and rearranging the ratio constraints we obtain

$$\begin{array}{rl} \min & -x_1x_2x_3 \\ \text{s.t.} & (2/a_1)x_1x_2 + (2/a_1)x_1x_3 + (2/a_1)x_2x_3 \le 1, \\ & (1/a_2)x_2x_3 \le 1, \\ & r_1(x_1)^{-1}x_2 \le 1, \\ & (1/r_2)x_1(x_2)^{-1} \le 1, \\ & r_3(x_2)^{-1}x_3 \le 1, \\ & (1/r_4)x_2(x_3)^{-1} \le 1, \\ & x_1, x_2, x_3 > 0, \end{array}$$

which is a geometric program.

2 Compressed Sensing

We want to solve the following optimization problem.

$$\begin{array}{rl} \min & \|x\|_1 \\ \text{s.t.} & Ax = y, \\ & x \in \mathbb{R}^n. \end{array}$$

The norm in the objective function is given by the sum of the absolute values of the vector components. To convert the problem into standard form we introduce the following auxiliary variables.

$$x_i^+ = \max\{x_i, 0\}, \qquad x_i^- = -\min\{x_i, 0\}$$

Then we have

$$x_i = x_i^+ - x_i^-, \qquad |x_i| = x_i^+ + x_i^-.$$

If we now let $\tilde{x} = (x_1^+, \dots, x_n^+, x_1^-, \dots, x_n^-)^T$, and $B = [A, -A]$, then we have an equivalent formulation in standard form:

$$\begin{array}{rl} \min & \displaystyle\sum_{i=1}^{2n} \tilde{x}_i \\ \text{s.t.} & B\tilde{x} = y, \\ & \tilde{x}_i \ge 0. \end{array}$$

Running the compressed sensing algorithm five times each for different numbers of non-zero signal entries s, and different numbers of measurements m we obtained Figure 1, which displays the average success rate given the parameters.

The curve plotted in blue through the figure is given by

$$m = s(1 + \log(n/s)).$$

We can see a very clear pattern in that there is a very high chance of success if the number of measurements is greater than this estimate, and a very low chance of success if it is lower. In between the two regions there is a rather thin transition region.

We can observe that the number of measurements should always be higher than the number of non-zero signal entries, but only by the small factor of $1 + \log(n/s)$.

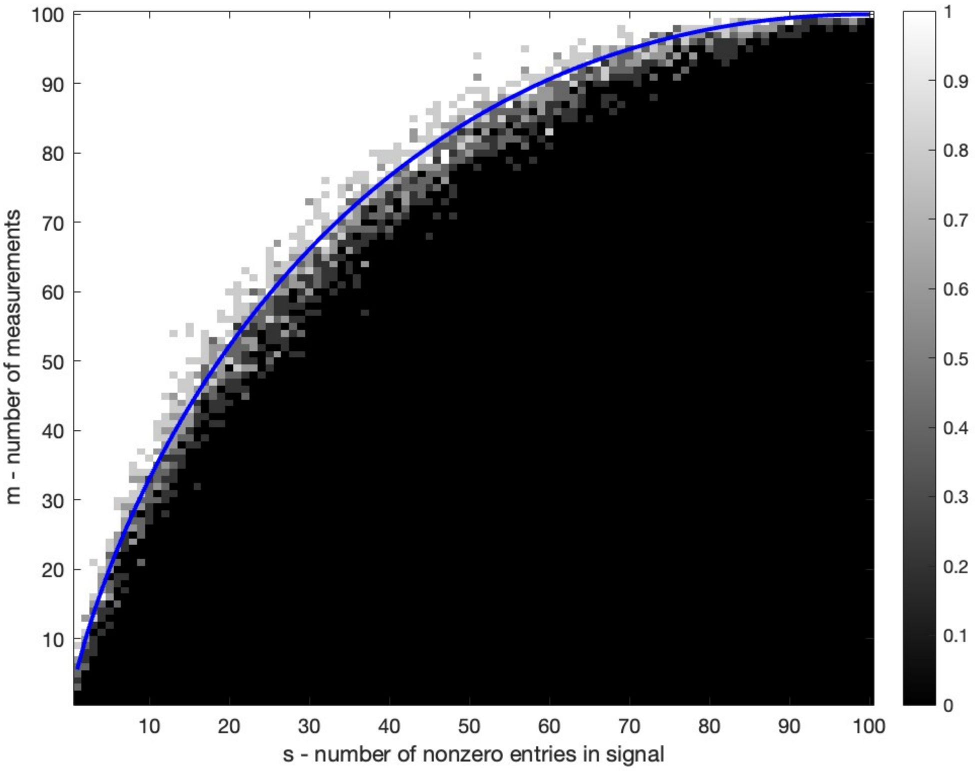

Figure 1: Success rates of compressed sensing along with the curve $m = s(1 + \log(100/s))$.

3 Markowitz Portfolio Selection

The results of portfolio selection problem are presented here. Figure 2 shows the expected return plotted against the risk tolerance parameter R.

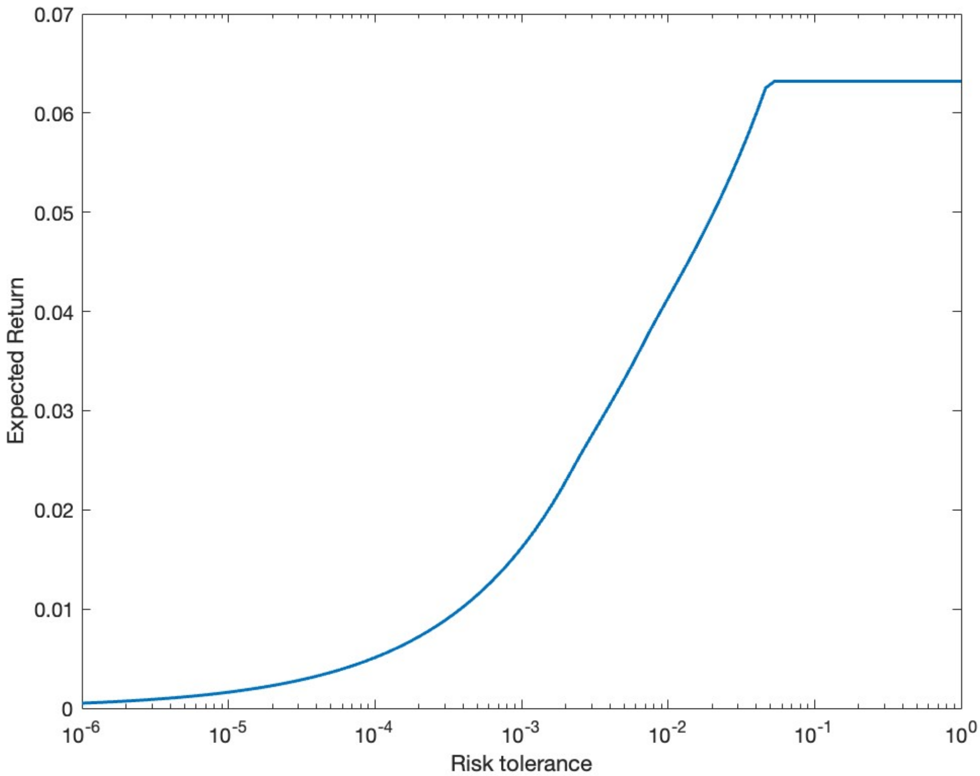

Figure 2: Expected returns of optimal portfolio strategies satisfying different risk tolerance constraints.

We can see that the expected return goes up as more risk is tolerated. This is a well-known result. The upper bound can be explained by studying the actual portfolio strategies selected by the algorithm. Four of these are given in Table 1.

Stock	**AMR**	**1e-0**	**1e-2**	**1e-4**	**1e-6**
TSLA	6.3%	100%	35.6%	1.5%	0.15%
NVDA	3.5%	-	8.6%	1.6%	0.16%
LLY	2.8%	-	55.8%	7.2%	0.72%
COST	1.9%	-	-	0.6%	0.06%
MRK	1.7%	-	-	1.8%	0.18%
ABBV	1.6%	-	-	1.0%	0.10%
PG	1.3%	-	-	7.7%	0.77%
Total		100%	100%	21.4%	2.14%

Table 1: Selected optimal portfolio strategies.

Now, it is clear that the ceiling on the expected return comes from the strategy of allocating all of the funds to the best performing stock. As the risk tolerance decreases, other stocks have to be introduced to reduce the risk. At some point, the risk constraint becomes so binding, that not all the funds can be allocated anymore. In fact, the total amount allocated decreases with the square root of the risk tolerance, as the latter is related to the covariance matrix of the data.

4 Matrix Completion

The matrix completion instance is given as follows.

$$\begin{aligned} \min_X \quad & \|X\|_*, \\ \text{s.t.} \quad & X_{ij} = Z_{ij} \qquad \forall (i,j) \in \Omega. \end{aligned}$$

With the given expression for the nuclear norm we have

$$\begin{aligned} \min_X \quad & \frac{1}{2} \min \left\{ \mathrm{tr}(W_1) + \mathrm{tr}(W_2) \mid \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix} \text{is positive semi-definite.} \right\}, \\ \text{s.t.} \quad & X_{ij} = Z_{ij} \qquad \forall (i,j) \in \Omega. \end{aligned}$$

We can simplify this instance in several ways. First, the factor 1/2 in the objective function can be safely ignored. Second, the nested minimum operators can be combined into one, by taking the conjunction of the constraints. Third, we recognize that the sum of the trace of W_1 and the trace of W_2 is equal to the trace of the matrix above. Therefore, we have

$$\begin{aligned} \min_X \quad & \mathrm{tr} \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix}, \\ \text{s.t.} \quad & \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix} \text{is positive semi-definite.}, \\ & X_{ij} = Z_{ij} \qquad \forall (i,j) \in \Omega. \end{aligned}$$

This is not quite in semi-definite form yet. The decision variable in this instance is still X. If we let B denote the matrix above, however, we can let the decision variable be B instead, since W_1 and W_2 are free. Finally, we simply have to express the equality constraints as matrix trace products. Let E_{ij} denote the matrix with all zeros except for a one in the i,j-th entry, and let m denote the number of rows of X. Then the above instance is equivalent to the following semi-definite program.

$$\begin{aligned} \min_B \quad & \mathrm{tr}(IB), \\ \text{s.t.} \quad & \mathrm{tr}(E_{i,m+j}B) = Z_{ij} \qquad \forall (i,j) \in \Omega, \\ & B \text{ is positive semi-definite.} \end{aligned}$$

Solving the matrix completion problem for different sizes of the known index set Ω we obtain Figure 3. We can see that about 1,800 out of 10,000 entries should be known for an underlying rank two structure to be approximated with minimal error.

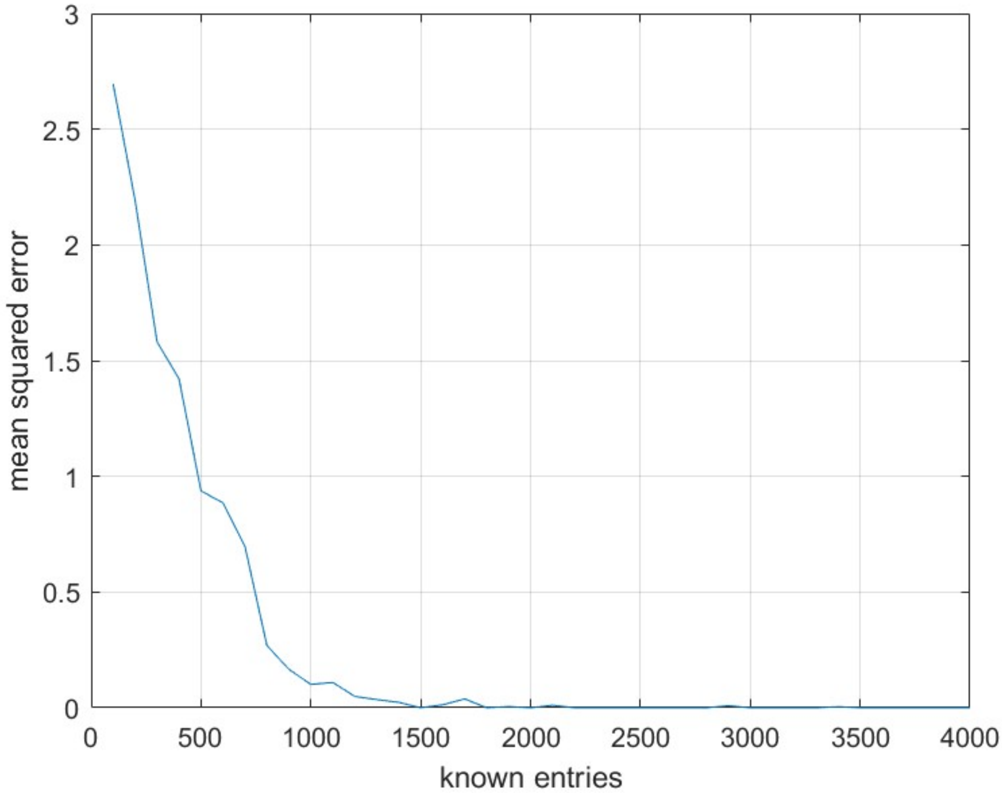

Figure 3: Mean squared error by the amount of known entries out of 10,000.

5 Polynomial Optimization

For the first polynomial p we indeed get the optimal value $\gamma = 4$. For p_1 and p_2 we get the optimal values $\gamma_1 = -1$ and $\gamma_2 = 0$ respectively. From the obtained Cholesky decompositions of the positive semi-definite matrices found for those γ values result the following sum of squares formulas, where the coefficients have been rounded to two decimal places.

$$\begin{aligned}
p(x) - 4 &= x^4 + 4x^3 + 6x^2 + 4x + 1 \\
&= (1 + 2x + x^2)^2 + (0.1x + 0.1x^2)^2 + (0.01x^2)^2, \\
p_1(x) - 1 &= x^6 + 4x^4 + 10x^3 + 4x^2 + 1 \\
&= (1 - 0.4x^2 + 0.6x^3)^2 + (2.19x + 2.01x^2 - 0.18x^3)^2 + (0.78x^2 + 0.78x^3)^2, \\
p_2(x) &= x^6 + 4x^4 + 10x^3 + 6x^2 \\
&= (-1.39x^2 - 0.81x^3)^2 + (2.45x + 2.04x^2 - 0.47x^3)^2 + (0.47x^2 - 0.32x^3)^2 + (0.14x^3)^2.
\end{aligned}$$