Homework 9
High-Dimensional Probability for Data Science, Fall 2023

Hints are in the back of the homework set.

The first problem is about an arbitrary set of n unit vectors in $\mathbb{R}^n$, i.e. vectors $x_1, \ldots, x_n$ satisfying $\|x_i\|_2 = 1$ for all i. Their sum $x_1 + \cdots + x_n$ has norm at most n, due to the triangle inequality. The bound n is obviously optimal, and is attained if all vectors v_i are the same. However, the sum can be made much smaller by carefully selecting the *signs* for the vectors x_i.

1. Balancing vectors

Let $x_1, \ldots, x_n$ be an arbitrary set of unit vectors in $\mathbb{R}^n$. Prove that there exist $\varepsilon_1, \ldots, \varepsilon_n \in \{-1, 1\}$ such that

$$\|\varepsilon_1 x_1 + \cdots + \varepsilon_n x_n\|_2 \leq \sqrt{n}.$$

As we learned in class, the *kernel method* in machine learning consists of replacing the inner product $\langle x, y \rangle$ on $\mathbb{R}^d$ with a suitable function $K(x, y)$. Not all functions are allowed, and Mercer's theorem (Lecture 16) gives a necessary and sufficient condition: K must be a *kernel*. Recall that a kernel is a function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that, for any number $n \in \mathbb{N}$ and vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, the $n \times n$ matrix $\Big[K(x_i, x_j)\Big]_{i,j=1}^n$ is symmetric and positive semidefinite.

Mercer's condition is not always convenient to check. It might be simpler to build kernels from "building blocks": simple kernels. Here you will verify some building rules.

2. How to build a kernel

Let $K(x, y)$ and $M(x, y)$ be kernels. Show that all of the following are kernels, too:

(a) $aK(x, y) + bM(x, y)$, where $a, b > 0$ are constants;

(b) $K(x, y)^p$ where $p \in \mathbb{N}$;

(c) $P(K(x, y))$ where P is a polynomial with nonnegative coefficients.

A low-rank matrix is an analog of low-dimensional data in the matrix world. It is therefore not surprising that many algorithms in linear algebra work faster for low-rank matrices. So it may be wise to replace a given matrix A by its best low-rank approximation A_k before we put it into a numerical algorithm. How well can one approximate an arbitrary matrix by a low-rank matrix? In this problem, we will prove bounds that are optimal in general.

3. Low-rank approximation

Consider an arbitrary $n \times m$ matrix A. Let $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_r$ denote the singular values of A, where $r = \min(n, m)$. Let $k \leq r$ be any nonnegative integer.

(a) Show that there exists a matrix A_k of rank k such that

$$\|A - A_k\| \leq \sigma_{k+1}(A).$$

(b) Show that there exists a matrix A_k of rank k such that

$$\|A - A_k\| \leq \frac{1}{\sqrt{k+1}} \|A\|_F .$$

---

4. Semidefinite relaxations of combinatorial problems

In Lecture 16, we studied the following NP-hard combinatorial problem. Given an $n \times m$ matrix $A = (a_{ij})$, find a maximum of the function

$$f(x, y) = \sum_{i,j} a_{ij} x_i y_j$$

over all sign vectors $x = (x_1, \ldots, x_n) \in \{-1, 1\}^n$ and $y = (y_1, \ldots, y_m) \in \{-1, 1\}^m$.

This problem can be approximately relaxed to a semidefinite program, which can always be solved in polynomial time. In class, we did this only for $n = m$ and $x_i = y_i$; now we will work in full generality.

Consider maximizing the function

$$g(U, V) = \sum_{i,j} a_{ij} \langle u_i, v_j \rangle$$

over all choices of unit vectors u_i and v_j in $\mathbb{R}^d$.

(a) (Accuracy) Show that for every d, the maximum of $g(U, V)$ is within a constant factor of the the maximum of the maximum of $f(x, y)$.

(b) (Efficiency) Choose $d = n + m$ and express maximizing $g(U, V)$ as a semidefinite program.[1]

TURN OVER FOR HINTS

[1] A general form of a semidefinite program is the following: maximize $g(Z)$ over the set of positive semidefinite matrices Z satisfying $h_i(Z) = b_i$ for all $i = 1, \ldots, N$. Here g and h_i are given linear functions from $\mathbb{R}^{d \times d} \to \mathbb{R}$, and b_i are given real numbers. There exist many commercial and open-source solvers for programs of this type.

Hints

Hints for Problem 1.

Make use of a probabilistic method. Choose ε_i at random and independently, and compute $\mathbb{E}\|\varepsilon_1 x_1 + \cdots + \varepsilon_n x_n\|_2^2$. Additivity of variance can help (Homework 2, Problem 4).

Hints for Problem 2.

You may use without proof any and all standard facts about positive semidefinite matrices. Part (c) should follow from (a) and (b).

Hints for Problem 3.

(a) Consider a truncated singular value decomposition $A_k = \sum_{i>k} \sigma_i u_i v_i^\mathsf{T}$.

(b) Express the Frobenius norm of A via the singular values, then use part (a). This will reduce the problem to proving the inequality $s_p \le \frac{1}{p}\sum_{i=1}^r s_i$ for any sequence of nonincreasing numbers s_i and any $p \le r$. (In our case, $s_i = \sigma_i(A)^2$ and $p = k+1$.) The inequality can be proved e.g. by contradiction.

Hints for Problem 4.

(a) Recall Grothendieck's inequality.

(b) Let U be $n \times d$ matrix whose rows are u_i^T, and let V be $m \times d$ matrix whose rows are v_j^T. Note that $f(x,y) = x^\mathsf{T} A y$, and check that a similar inequality holds for g, namely $g(U,V) = \operatorname{tr}(U^\mathsf{T} A V)$. Next, use the following so-called "hermitization trick": consider the $d \times d$ symmetric matrix $H = \begin{bmatrix} 0 & A \\ A^\mathsf{T} & 0 \end{bmatrix}$ and the $d \times d$ matrix $R = \begin{bmatrix} U \\ V \end{bmatrix}$. Check that $R^\mathsf{T} H R = U^\mathsf{T} A V + V^\mathsf{T} A^\mathsf{T} U$, and conclude that $g(U,V) = \frac{1}{2}\operatorname{tr}(R^\mathsf{T} H R) = \frac{1}{2}\operatorname{tr}(H R R^\mathsf{T})$. Now since in this problem u_i and v_j are arbitrary unit vectors, $Z = RR^\mathsf{T}$ is a Gram matrix of any system of $n+m$ unit vectors in $\mathbb{R}^d$. Thus Z is an arbitrary positive semidefinite matrix with unit diagonal entries (why?). This will allow you to express the problem as maximizing $\frac{1}{2}\operatorname{tr}(HZ)$ over all positive semidefinite matrices Z with unit diagonal entries.