PSET 1

Due on **Friday September 20, 2023 at 11:59 pm ET**

COMPSCI 2243: Algorithms for Data Science, Fall 2024 (Sitan Chen) Version: **1.0**

Submitted by **<name>**

This problem set will cover concepts from the first unit on tensor decomposition.
The three questions have been labeled with the date of the lecture in which the relevant material is covered, to help you budget your time. The questions are meant to be challenging, so as with pset 0, do not feel discouraged if you get stuck and are unable to solve some of them. If you find that you are running low on time to finish all the problems, our recommendation is to try to aim for breadth rather than depth – e.g., it is better to complete a few parts of each of the three questions, than to completely solve one of the three questions and skip the others.

1 (50 PTS.) FUN WITH MOMENTS AND TENSORS (9/9)

Motivation: In class we saw a few examples where, by manipulating the moments of some probability distribution, we produced tensors whose components correspond to the parameters of that distribution, at which point we could use Jennrich's algorithm to learn those parameters. In this exercise, we will explore two toy applications of tensor decomposition to help you become comfortable with designing tensors using such moment manipulations.

1.A. (25 PTS.*) [Learning a low-rank polynomial]

Setup: Let $v_1, \dots, v_d \in \mathbb{R}^d$. You may assume that $d \geqslant 4$. Let $V \in \mathbb{R}^{d \times d}$ be the matrix whose rows consist of $v_1, \dots, v_d$, and suppose that V is invertible.

Suppose we are given many pairs of the form (x, y) where $x \sim \mathcal{N}(0, \mathbb{1}_d)$ and

$$y = \sum_{i=1}^{d} \langle v_i, x \rangle^3 .$$

Question: Give an algorithm based on tensor decomposition that can learn $v_1, \dots, v_d$ to small error. The algorithm should involve estimating various moments of the joint distribution on (x, y), i.e. quantities of the form $\mathbf{E}[p(x, y)]$ for various choices of polynomial $p : \mathbb{R}^{d+1} \to \mathbb{R}$. You may assume in your proof of correctness that you can compute these moments exactly.

(Note: this can be solved just by using polynomial regression, but this exercise is asking for a different algorithm. While the latter might seem like an odd choice in the setting above, we will see later in this course how to use ideas based on this alternative algorithm to prove results that *cannot* be obtained through polynomial regression.)

(**Hint:** *Consider polynomials of the form* $p(x, y) = y \cdot \prod_{j \in S} x_j$ *for various choices of subsets* $S \subseteq [d]$ *of size at most* 3.)

1.B. (25 PTS.*) [Learning a simple nonlinear pushforward]

Setup: Let $v_1, \dots, v_d$ and V be as in the previous question. Consider a ***pushforward distribution*** q over $\mathbb{R}^d$ defined as follows. To sample from q, one samples a fresh vector $g \sim \mathcal{N}(0, \mathbb{1}_d)$ and outputs the vector $w \in \mathbb{R}^d$ whose j-th coordinate is given by

$$w_j = \sum_{i=1}^{d} (v_j)_i \cdot g_i^2 .$$

Question: Give an algorithm based on tensor decomposition that, given samples from q, can learn the columns of V up to permutation. The algorithm should involve estimating various moments of q, and as in Part **1.A.**, you may assume in your proof of correctness that you can compute these moments exactly.

Solution:

1.A.

1.B.

2 (50 pts.) Condition number and noise robustness of eigendecomposition (9/9)

Motivation: In the first two lectures we saw two closely related algorithms, matrix pencil method and Jennrich's algorithm, that both rely on computing the eigendecomposition of some matrix. In lecture, these algorithms were presented under the assumption that we had exact access to the matrix when, in applications of these algorithms, we only have access to the matrix plus some additive noise. In this problem, you will show that such eigendecompositions are "robust" to this noise.

Setup: Let A be an $n \times n$ diagonalizable matrix with eigendecomposition $A = Q\Lambda Q^{-1}$ where $\lambda_i = \Lambda_{ii}$ for each $i \in [n]$ are the eigenvalues of A and the columns $q_1, \ldots, q_n$ of Q are unit vectors corresponding to eigenvectors of A. Define $\delta := \min_{i \neq j} |\lambda_i - \lambda_j|$. Let

$$\widetilde{A} = A + \Delta\,,$$

where Δ is an $n \times n$ matrix and represents noise. Our goal will be to establish that the eigenvalues and eigenvectors of $\widetilde{A}$ are close to those of A, provided three conditions hold:

(a) $\|\Delta\|$ is sufficiently small,

(b) The eigenvalues of A are sufficiently distinct, i.e. δ is not too small,

(c) Q is "robustly full rank."

Aside: We first formalize what it means to be robustly full rank. Let $\kappa(Q) = \|Q\|/\sigma_{\min}(Q)$ denote the ***condition number*** of Q, where $\|Q\|$ is the operator norm of Q and $\sigma_{\min}(Q)$ is the minimum singular value of Q. Note that if Q is not full rank, then condition number is infinite. More generally, if Q is "close" to being full rank in the sense that there is a linear combination of columns of Q which has small norm, where the coefficients are not too small, then $\kappa(Q)$ is large. If $\kappa(Q)$ is small, then we think of Q as being "robustly full rank." Note that $\kappa(Q)$ is scale-invariant: for any $\alpha > 0$, $\kappa(\alpha \cdot Q) = \kappa(Q)$.

2.A. (5 pts.) Let $\widetilde{\Delta} = Q^{-1}\Delta Q$. Verify that

$$Q^{-1}\widetilde{A}Q = \Lambda + \widetilde{\Delta}\,.$$

Then show that

$$\|\widetilde{\Delta}\|_{\max} \leqslant \kappa(Q)\|\Delta\|\,.$$

In the next question you may use the following theorem:

Theorem 2.1 (Gershgorin's disk theorem). *Let $M \in \mathbb{R}^{n\times n}$. All eigenvalues of M lie in the union of disks $\cup_i C_i$ where*

$$C_i \triangleq \{z \in \mathbb{C} : \|z - M_{ii}\| \leqslant \rho_i\} \quad \text{for} \quad \rho_i \triangleq \sum_{j:j\neq i} |M_{ij}|.$$

2.B. (15 pts.) Use Gershgorin's disk theorem to show that when

$$r := \max_i \sum_j |\widetilde{\Delta}_{ij}| < \delta/2\,,$$

then $\widetilde{A}$ has distinct eigenvalues $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$, and furthermore

$$|\lambda_i - \widetilde{\lambda}_i| < \delta/2 \tag{1}$$

for all $i \in [n]$.

2.C. (2 pts.) Conclude from Parts **2.A.** and **2.B.** that $\widetilde{A}$ is diagonalizable when

$$\|\Delta\| < \frac{\delta}{2\kappa(Q)n} \tag{2}$$

and has eigenvalues $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$ satisfying Eq. (1).

2.D. (18 pts.*) Assume (2) holds and let $\widetilde{A} = \widetilde{Q}\widetilde{\Lambda}\widetilde{Q}^{-1}$ be the eigendecomposition of $\widetilde{A}$ where the columns $\widetilde{q}_1, \ldots, \widetilde{q}_n$ of $\widetilde{Q}$ are unit vectors corresponding to eigenvectors of $\widetilde{Q}$ with eigenvalues $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$ respectively. We will now argue that the eigenvectors of $\widetilde{A}$ are close to those of A.
To formalize this, let W be an $n \times n$ matrix such that $\widetilde{q}_i = \sum_j W_{ij} q_j$ for each $i \in [n]$. Show

$$|W_{ij}| \leqslant \frac{2\|\Delta\|}{\delta \cdot \sigma_{\min}(Q)}$$

for all $i \neq j$.
(**Hint:** *Write $\widetilde{A}\widetilde{q}_i$ in two different ways, one by using $\widetilde{A} = A + \Delta$, and the other by using $\widetilde{q}_i = \sum_j W_{ij} q_j$. You may also find it helpful to use the rows of Q^{-1}.*)

2.E. (10 PTS.) Show that for all $i \in [n]$, there is some $\sigma_i \in \mathbb{C}$ satisfying $|\sigma_i| = 1$ such that

$$\|q_i - \sigma_i \widetilde{q}_i\| \leqslant O\Big(\frac{n\|\Delta\|}{\delta \cdot \sigma_{\min}(Q)}\Big)$$

for every $i \in [n]$. This means that if the noise is relatively small then $\widetilde{Q}$ is close to Q.

Solution:

2.A.

2.B.

2.C.

2.D.

2.E.

3 (50 pts.) Landscape analysis for orthogonal tensor decomposition (9/11)

Motivation: In class we saw an analysis of tensor power method for orthogonal tensor decomposition, as well as a formal connection between tensor power method and gradient descent. In this exercise, we explore an alternative approach to orthogonal tensor decomposition: directly analyzing the optimization landscape in order to show that gradient descent converges to a component of the tensor.

Setup: Throughout, let

$$T = \sum_{i=1}^{d} \lambda_i u_i^{\otimes 3}$$

for $u_1, \ldots, u_d \in \mathbb{R}^d$ a collection of orthonormal vectors and $\lambda_1, \ldots, \lambda_d > 0$. Let $p : \mathbb{R}^d \to \mathbb{R}$ be the associated cubic polynomial

$$p(x) = \sum_i \lambda_i \langle u_i, x \rangle^3 .$$

The result we will prove is the following:

Theorem 3.1. *Let $x \in \mathbb{S}^{d-1}$ be any point for which $p(x) \geqslant 0$. Then x is a* ***strict local maximum*** *for p over the domain $\mathbb{S}^{d-1}$ if and only if $x = u_i$.*[1]

In other words, there are no "spurious local maxima" for the objective function p, which means that we can find a component u_i of the tensor T simply by running an appropriate implementation of gradient descent.[2]

3.A. (2 pts.) It suffices to establish Theorem 3.1 when $u_1, \ldots, u_d$ are the standard basis vectors, i.e. when $u_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 entry is in the i-th coordinate. Give an informal argument for why this is the case.

For the remaining questions in this exercise, we will assume that $u_1, \ldots, u_d$ are the standard basis vectors.

3.B. (10 pts.) Prove that every u_i is a strict local maximum.

3.C. (3 pts.) Recall that any local maximum x of p over $\mathbb{S}^{d-1}$ is a ***stationary point***, that is, the projection of the gradient of p at x to the tangent space at x is zero. Also recall that the projector to the tangent space at x is the operator $\mathbb{1} - xx^\top$. With this in mind, prove that any local maximum x satisfies

$$\lambda_i x_i^2 = p(x) \cdot x_i$$

for all $i \in [d]$.

3.D. (10 pts.) Prove that if $x \in \mathbb{S}^{d-1}$ satisfies $p(x) = 0$, then x is not a strict local maximum.
(**Hint:** *Consider the effect of perturbing one of the coordinates of x by a small amount and rescaling so that the vector has unit norm. Alternatively, use Part* **3.C.**.)

We now arrive at the trickiest part of the proof. Let $x \in \mathbb{S}^{d-1}$ be any point that satisfies $p(x) > 0$ and has more than one nonzero coordinate. We need to show that such an x cannot be a local maximum. Let $S^* \subset [d]$ denote the subset of coordinates of x which are nonzero.

3.E. (5 pts.) Let S be any proper subset of S^*. Construct a unit vector w such that 1) $\langle w, x \rangle = 0$, 2) the entries of w indexed by S are given by scaling all the entries of x indexed by S by the same factor, and 3) the entries of w indexed by $[d] \backslash S$ are given by scaling all the entries of x indexed by $[d] \backslash S$ by the same factor.

3.F. (20 pts.*) Prove that x is not a strict local maximum.
(**Hint:** *Perturb x in the direction of w and rescale it to give a unit vector $x' = \delta w + \sqrt{1 - \delta^2} x$. Argue that p achieves a higher value at x' than at x. You may find it helpful to Taylor expand $p(x')$ around $\sqrt{1 - \delta^2} x$ and apply Part* **3.C.**)

Solution:

3.A.

3.B.

3.C.

3.D.

3.E.

3.F.

[1] Recall that a point $x \in \mathbb{S}^{d-1}$ is said to be a ***strict local maximum*** for a function $f : \mathbb{S}^{d-1} \to \mathbb{R}$ if there exists some $\epsilon > 0$ such that for all $x' \in \mathbb{S}^{d-1}$ which are distinct from x and which satisfy $\|x' - x\|_2 \leqslant \epsilon$, we have $f(x') < f(x)$.

[2] There are some nontrivial subtleties regarding escaping from saddle points that we will not cover in this course. If you are interested, one of the canonical references is this paper.