Math 126 Problem 16: Efficiently Computing the SVD

Spring 2027

Submission Instructions

Please upload your work to Gradescope by **11:59pm on Monday, April 5, 2027 (Pacific time).**

- You are encouraged to discuss ideas and work with your classmates. However, you *must acknowledge* your collaborators at the top of each solution on which you collaborated with others and you *must write* your solutions independently.
- Your solutions to theory questions must be written legibly, or typeset in LaTeX or markdown.
- I recommend that you write your solutions in a Python notebook using Google Colab.
- You should submit your solutions as a *single PDF* via the assignment on Gradescope.

Grading: The point of the problem set is for *you* to learn. To this end, I hope to disincentivize the use of LLMs by *not* grading your work for correctness. Instead, you will grade your own work by comparing it to my solutions. This self-grade is due the Friday *after* the problem set is due, also on Gradescope.
Original notebook: Google Colab.

Efficiently Computing the SVD

We will explore how to efficiently approximate the top singular vector of a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. Instead of computing the full SVD (which takes $O(nd^2)$ time), we will analyze and implement two iterative algorithms operating on the symmetric matrix $\mathbf{A} = \mathbf{X}^\top \mathbf{X}$: the Power Method and the Lanczos Method.

Part A: Tridiagonal Recurrence

The efficiency of the Lanczos method relies entirely on the fact that projecting a symmetric matrix $\mathbf{A}$ onto its Krylov subspace results in a tridiagonal matrix. In this part, you will prove why this happens.

Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be a symmetric matrix. Let $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_q]$ be a matrix with orthonormal columns that span the degree q Krylov subspace generated by $\mathbf{A}$ and an initial vector $\mathbf{z}^{(0)}$. Let $\mathbf{T} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q}$.

1. **The Hessenberg Property**: By the definition of the Krylov subspace, we know that $\mathbf{A}\mathbf{q}_j$ is strictly contained within the span of the first $j+1$ basis vectors. Use this fact to prove that $T_{i,j} = 0$ for all $i > j + 1$.

2. **Symmetry**: Prove that the matrix $\mathbf{T}$ is symmetric ($\mathbf{T} = \mathbf{T}^\top$).

Combine your results from steps 1 and 2 to rigorously conclude that $T_{i,j} = 0$ for all $|i - j| > 1$. What does this tell us about the structure of $\mathbf{T}$?

Part B: Power Method Algorithm

Write a Python function to implement the Power Method. Your function should take a symmetric matrix $\mathbf{A}$ and the number of iterations q, and return the approximated top eigenvector.

Part C: Lanczos Algorithm

Now, write a Python function to implement the Lanczos Method. Remember that in practice, we iteratively build $\mathbf{Q}$ and $\mathbf{T}$ using the three-term recurrence to ensure numerical stability and speed.

Hint: For finding the top eigenvector of your small $q \times q$ matrix $\mathbf{T}$, you may use `np.linalg.eigh`.

Part D: Comparison

We will test these algorithms on the digits dataset from `scikit-learn`. Extract the data matrix $\mathbf{X}$ and mean-center its columns to form the covariance matrix $\mathbf{A} = \mathbf{X}^\top \mathbf{X}$.

Run both your power method and Lanczos method implementations on $\mathbf{A}$ for $q = 20$ iterations. Plot the ℓ_2 distance between the estimated true top eigenvector by iteration. What do you notice?