CSCI 145 Problem 3 Solution: The Cost of a Matrix Power

Solution

Approach

We count the inner products required by each strategy. Applying $\mathbf{A}$ repeatedly avoids matrix-matrix products, while repeated squaring reduces their number from linear to logarithmic in q. For a rank-one matrix, we can avoid matrix-matrix products entirely.

1. Forming the Matrix Power First

Forming $\mathbf{A}^q$ by the stated sequence requires $q-1$ dense matrix-matrix multiplications. Each output has d^2 entries, and each entry is a length-d inner product, so one matrix-matrix multiplication costs $O(d^3)$. Multiplying the completed matrix by $\mathbf{x}$ then costs $O(d^2)$. The total is therefore

$$O\big((q-1)d^3 + d^2\big) = O(qd^3).$$

2. Applying Matrix-Vector Products First

One dense matrix-vector product has d entries, each computed by a length-d inner product, so it costs $O(d^2)$. Applying $\mathbf{A}$ to the current vector q times costs

$$O(qd^2).$$

3. Repeated Squaring

Let $L = \lfloor \log_2 q \rfloor$ and write the binary expansion of q as

$$q = \sum_{j=0}^{L} b_j 2^j, \qquad b_j \in \{0, 1\}.$$

Repeated squaring constructs the powers needed by this expansion:

$$\mathbf{A}^2, \quad \mathbf{A}^4 = (\mathbf{A}^2)^2, \quad \mathbf{A}^8 = (\mathbf{A}^4)^2, \quad \ldots \quad , \mathbf{A}^{2^L}.$$

This takes L dense matrix-matrix multiplications. Multiply the powers for which $b_j = 1$ to form $\mathbf{A}^q$. They give the desired exponent because

$$\prod_{j:b_j=1} \mathbf{A}^{2^j} = \mathbf{A}^{\sum_j b_j 2^j} = \mathbf{A}^q.$$

The selected factors may be multiplied in any order because they are all powers of the same matrix. There are at most $L+1$ selected powers, so forming their product also takes $O(\log q)$ matrix-matrix multiplications. The final multiplication by $\mathbf{x}$ costs $O(d^2)$. The total is

$$O(d^3 \log_2 q + d^2) = O(d^3 \log_2 q).$$

4. Using the Rank-One Factors

For $\mathbf{A} = \mathbf{u}\mathbf{v}^\top$, applying the matrix to any vector $\mathbf{y}$ gives

$$\mathbf{A}\mathbf{y} = \mathbf{u}(\mathbf{v}^\top \mathbf{y}).$$

The first two powers applied to $\mathbf{x}$ are therefore

$$\mathbf{A}\mathbf{x} = \mathbf{u}(\mathbf{v}^\top \mathbf{x}),$$
$$\mathbf{A}^2\mathbf{x} = \mathbf{u}(\mathbf{v}^\top \mathbf{u})(\mathbf{v}^\top \mathbf{x}).$$

Each additional application of $\mathbf{A}$ contributes one more factor of $\mathbf{v}^\top \mathbf{u}$, so

$$\mathbf{A}^q\mathbf{x} = (\mathbf{v}^\top \mathbf{u})^{q-1}\mathbf{u}(\mathbf{v}^\top \mathbf{x}).$$

Computing the two inner products costs $O(d)$, raising the scalar $\mathbf{v}^\top \mathbf{u}$ to the power $q-1$ by repeated squaring costs $O(\log q)$, and scaling $\mathbf{u}$ costs $O(d)$. The total is

$$O(d + \log q).$$

Repeated matrix-vector products cost $O(qd^2)$, while repeated squaring costs $O(d^3 \log q)$. Comparing these bounds, repeated squaring is faster when $d \log q$ is smaller than q, up to constant factors. For small q or large d, repeated matrix-vector products can cost less.