Deck 3: Matrix multiplication: preasymptotic estimation

Math 7870: Topics in Randomized Numerical Linear Algebra

Spring 2026

Akil Narayan

Recall: matmat

We proposed a randomized algorithm for approximating $\boldsymbol{AB}$ using *uniform sampling*.

The basic idea was to write $\boldsymbol{AB}$ as a sum of rank-1 outer products, and form an (unbiased) estimator by uniformly at random summing N of the rank-1 matrices.

We identified, in principle, the type of distribution that the estimator has: by the CLT, a normal centered random variable with a total variance scaling like $1/N$.

Recall: matmat

We proposed a randomized algorithm for approximating $\boldsymbol{AB}$ using *uniform sampling*.

The basic idea was to write $\boldsymbol{AB}$ as a sum of rank-1 outer products, and form an (unbiased) estimator by uniformly at random summing N of the rank-1 matrices.

We identified, in principle, the type of distribution that the estimator has: by the CLT, a normal centered random variable with a total variance scaling like $1/N$.

What needs to be done: guarantees, and pre-asymptotic estimation.

A simplification of matmat

It'll be convenient for us to get the crux of the ideas by simplifying the problem:

Given vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$, let's approximate $\langle \boldsymbol{b}, \boldsymbol{a} \rangle = \boldsymbol{a}^T \boldsymbol{b}$ using the same idea as before.

The goal is to *not* sample the entire set of entries of both vectors.

A simplification of matmat

It'll be convenient for us to get the crux of the ideas by simplifying the problem:

Given vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$, let's approximate $\langle \boldsymbol{b}, \boldsymbol{a} \rangle = \boldsymbol{a}^T \boldsymbol{b}$ using the same idea as before.

The goal is to *not* sample the entire set of entries of both vectors.

The procedure now is a little more transparent:

$$\boldsymbol{a}^T \boldsymbol{b} = \sum_{j \in [k]} a_j b_j \implies p_X(k a_j b_j) = \frac{1}{k},$$

so that,

$$\mathbb{E}X = \sum_{j \in [k]} p_X(k a_j b_j) k a_j b_j = \sum_{j \in [k]} a_j b_j = \boldsymbol{a}^T \boldsymbol{b}$$

E.g. $\mathrm{Var}\, X = \mathbb{E}X^2 - (\mathbb{E}X)^2 = \sum_j \frac{1}{k} k^2 |a_j b_j|^2 - |a^T b|^2 = k \sum_{j \in [k]} |a_j b_j|^2 - \left(\sum_{j \in [k]} a_j b_j\right)^2$

Concentration for inner products

We can explicitly compute,

$$\mathrm{Var}X = \mathbb{E}X^2 - (\mathbb{E}X)^2 = k\sum_{j\in[k]}(a_j b_j)^2 - \left(\sum_{j\in[k]} a_j b_j\right)^2.$$

Therefore, if $X_n \overset{\text{iid}}{\sim} X$, then by the LLN + CLT,

$$\lim_{N\uparrow\infty}\frac{1}{N}\sum_{n\in[N]} X_n = \boldsymbol{a}^T\boldsymbol{b}, \qquad \sqrt{N}\left(\frac{1}{N}\sum_{n\in[N]} X_n - \boldsymbol{a}^T\boldsymbol{b}\right) \overset{N\uparrow\infty}{\sim} \mathcal{N}(0, \mathrm{Var}(X)).$$

This is, again, only asymptotic.

Concentration for inner products

We can explicitly compute,

$$\mathrm{Var}X = \mathbb{E}X^2 - (\mathbb{E}X)^2 = k\sum_{j\in[k]}(a_j b_j)^2 - \left(\sum_{j\in[k]} a_j b_j\right)^2.$$

Therefore, if $X_n \overset{\text{iid}}{\sim} X$, then by the LLN + CLT,

$$\lim_{N\uparrow\infty}\frac{1}{N}\sum_{n\in[N]} X_n = \boldsymbol{a}^T\boldsymbol{b}, \qquad \sqrt{N}\left(\frac{1}{N}\sum_{n\in[N]} X_n - \boldsymbol{a}^T\boldsymbol{b}\right) \overset{N\uparrow\infty}{\sim} \mathcal{N}(0, \mathrm{Var}(X)).$$

This is, again, only asymptotic.

However, we do have a preasymptotic quantitative understanding: $\mathrm{Var}\frac{1}{N}\sum_{n\in[N]} X_n = \frac{1}{N}\mathrm{Var}X$. We can compute this variance. Define a vector $\boldsymbol{c}$ as,

$$\boldsymbol{c} = \boldsymbol{a}\odot\boldsymbol{b} \in \mathbb{R}^k, \qquad c_j = a_j b_j, \qquad \mathbb{E}X = \mathbf{1}^T\boldsymbol{c}.$$

Then we have,

$$\mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T\boldsymbol{c}\right|^2$$

Best- and worst-case variance

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b}, \qquad \mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T \boldsymbol{c}\right|^2$$

Good algorithmic performance: $\mathrm{Var}X$ is small, relative to the (squared) oracle value.
What kinds of vectors $\boldsymbol{c}$ maximize/minimize the variance?

Best- and worst-case variance

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b}, \qquad \mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T \boldsymbol{c}\right|^2$$

Good algorithmic performance: $\mathrm{Var}X$ is small, relative to the (squared) oracle value.
What kinds of vectors $\boldsymbol{c}$ maximize/minimize the variance?

$$\frac{1}{\left|\mathbf{1}^T \boldsymbol{c}\right|^2}\mathrm{Var}X = \frac{k}{\left|\mathbf{1}^T \widehat{\boldsymbol{c}}\right|^2}, \qquad \widehat{\boldsymbol{c}} := \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|_2}.$$

Best- and worst-case variance

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b}, \qquad \mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T \boldsymbol{c}\right|^2$$

Good algorithmic performance: $\mathrm{Var}X$ is small, relative to the (squared) oracle value.
What kinds of vectors $\boldsymbol{c}$ maximize/minimize the variance?

$$\frac{1}{\left|\mathbf{1}^T \boldsymbol{c}\right|^2}\mathrm{Var}X = \frac{k}{\left|\mathbf{1}^T \widehat{\boldsymbol{c}}\right|^2} - 1, \qquad \widehat{\boldsymbol{c}} := \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|_2}.$$

The best case: $\widehat{\boldsymbol{c}} = \frac{1}{\sqrt{k}}\mathbf{1}$. Then $\mathrm{Var}X = 0$.
(I.e., each X_j takes a single value, equal to $\boldsymbol{a}^T\boldsymbol{b}$, with probability 1.)

Best- and worst-case variance

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b}, \qquad \mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T \boldsymbol{c}\right|^2$$

Good algorithmic performance: $\mathrm{Var}X$ is small, relative to the (squared) oracle value.
What kinds of vectors $\boldsymbol{c}$ maximize/minimize the variance?

$$\frac{1}{\left|\mathbf{1}^T \boldsymbol{c}\right|^2}\mathrm{Var}X = \frac{k}{\left|\mathbf{1}^T \widehat{\boldsymbol{c}}\right|^2}, \qquad \widehat{\boldsymbol{c}} := \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|_2}.$$

The best case: $\widehat{\boldsymbol{c}} = \frac{1}{\sqrt{k}}\mathbf{1}$. Then $\mathrm{Var}X = 0$.
(I.e., each X_j takes a single value, equal to $\boldsymbol{a}^T\boldsymbol{b}$, with probability 1.)

The worst case: $\boldsymbol{c} \perp \mathbf{1}$, i.e., $\boldsymbol{c}$ has positive and negative components of approximately the same mass. Then $\mathrm{Var}X = k\|\boldsymbol{c}\|_2^2$
(I.e., $\sum_n X_n$ sums positive and negative components with similar "mass".)

Best- and worst-case variance

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b}, \qquad \mathrm{Var}X = k\|\boldsymbol{c}\|_2^2 - \left|\mathbf{1}^T \boldsymbol{c}\right|^2$$

Good algorithmic performance: $\mathrm{Var}X$ is small, relative to the (squared) oracle value.
What kinds of vectors $\boldsymbol{c}$ maximize/minimize the variance?

$$\frac{1}{\left|\mathbf{1}^T \boldsymbol{c}\right|^2}\mathrm{Var}X = \frac{k}{\left|\mathbf{1}^T \widehat{\boldsymbol{c}}\right|^2}, \qquad \widehat{\boldsymbol{c}} := \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|_2}.$$

The best case: $\widehat{\boldsymbol{c}} = \frac{1}{\sqrt{k}}\mathbf{1}$. Then $\mathrm{Var}X = 0$.
(I.e., each X_j takes a single value, equal to $\boldsymbol{a}^T\boldsymbol{b}$, with probability 1.)

The worst case: $\boldsymbol{c} \perp \mathbf{1}$, i.e., $\boldsymbol{c}$ has positive and negative components of approximately the same mass. Then $\mathrm{Var}X = k\|\boldsymbol{c}\|_2^2$
(I.e., $\sum_n X_n$ sums positive and negative components with similar "mass".)

A near-worst case: $\widehat{\boldsymbol{c}} = \boldsymbol{e}_j$, so that $\mathrm{Var}X = (k-1)\|\boldsymbol{c}\|_2^2$.
(I.e., $\boldsymbol{a}^T\boldsymbol{b}$ has a bunch of zero summands, which we randomly sample with nonzero probability....)

Importance sampling, I

The near-worst case reveals a qualitative issue: sampling entries uniformly can provide suboptimal results.

An alternative: sampling based on knowledge of entries of $\boldsymbol{a}, \boldsymbol{b}$.

In particular, we can generalize our random variable to have a different mass function:

$$p_X\left(\frac{1}{p_j} a_j b_j\right) = p_j \text{ with } \sum_{j\in[k]} p_j = 1 \implies \mathbb{E}X = \boldsymbol{a}^T \boldsymbol{b}.$$

(Cf. $p_j = 1/k$)

($p_j > 0$)

We can craft the p_j values to improve performance. E.g., by minimizing variance.

$$\mathbb{E}X = \sum_j p_X\left(\frac{1}{p_j} a_j b_j\right) \cdot \frac{1}{p_j} a_j b_j = \sum_j p_j \frac{1}{p_j} a_j b_j = a^T b$$

Importance sampling, I

The near-worst case reveals a qualitative issue: sampling entries uniformly can provide suboptimal results.

An alternative: sampling based on knowledge of entries of $\boldsymbol{a}, \boldsymbol{b}$.

In particular, we can generalize our random variable to have a different mass function:

$$p_X\left(\frac{1}{p_j}a_j b_j\right) = p_j \text{ with } \sum_{j\in[k]} p_j = 1 \quad \Longrightarrow \quad \mathbb{E}X = \boldsymbol{a}^T\boldsymbol{b}.$$

We can craft the p_j values to improve performance. E.g., by minimizing variance.

Through a similar computation as before, we have,

$$\mathrm{Var}X = \mathbb{E}X^2 - (\mathbb{E}X)^2 = \sum_{j\in[k]} \frac{1}{p_j}(a_j b_j)^2 - \left(\sum_{j\in[k]} a_j b_j\right)^2 = \sum_{j\in[k]} \frac{1}{p_j} c_j^2 - |\mathbb{E}c|^2$$

So we can attempt to solve the problem:

$$\min_{p_j} \mathrm{Var}X \quad \text{subject to} \quad \sum_{j\in[k]} p_j = 1, \quad p_j > 0$$

(Lagrange multipliers)

$$\min_{p_j} \sum_j \frac{1}{p_j} c_j^2 \quad \text{s.t.} \quad \sum_{j \in [k]} p_j = 1, \quad p_j \geq 0 \ \forall j.$$

$p_j = s_j^2 \ \forall j$.

s_j: new "slack" variables, $s_j \neq 0$

$$\min_{\underline{p}, \underline{s}} f(\underline{p}, \underline{s}) \quad \text{s.t.} \quad g(\underline{p}, \underline{s}) = 0 \quad \text{and} \quad h_j(p_j, s_j) = 0 \quad \forall j \in [k]$$

Augmented Lagrangian: $\mathcal{L}(\underline{p}, \underline{s}) = f(\underline{p}, \underline{s}) + \lambda g(\underline{p}, \underline{s}) + \underline{\mu}^T \underline{h}(\underline{p}, \underline{s})$

Variables: $\underline{p}, \underline{s}, \lambda, \underline{\mu}$, compute stationary pts.

$$\frac{\partial \mathcal{L}}{\partial p_j} = -\frac{1}{p_j^2} c_j^2 + \lambda \cdot 1 + \mu_j \cdot 1 = 0$$

$$(h_j = p_j - s_j^2)$$

$$\frac{\partial \mathcal{L}}{\partial s_j} = -2 \mu_j s_j = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_j p_j - 1 = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mu_j} = p_j - s_j^2 = 0$$

$$s_j = \sqrt{p_j} \neq 0$$

$$\Rightarrow \mu_j = 0$$

$$\Rightarrow -\frac{1}{p_j^2} c_j^2 + \lambda = 0 \quad (1)$$

$$\sum_j p_j = 1 \quad (2)$$

$$(1): \frac{p_j^2}{c_j^2} = 1/\lambda \Rightarrow p_j = |c_j| \sqrt{1/\lambda}$$

$$(2): \sum_j p_j = 1 \Rightarrow \sqrt{1/\lambda} \left(\sum_j |c_j| \right) = 1$$

$$\sqrt{1/\lambda} = 1/\|\underline{c}\|_1$$

$$\Rightarrow p_j = |c_j| / \|\underline{c}\|_1$$

$$\mathrm{Var}\, X = \sum_j \frac{1}{p_j} |c_j|^2 - |\underline{1}^T \underline{c}|^2$$

$$= \|\underline{c}\|_1^2 - |\underline{1}^T \underline{c}|^2$$

Importance sampling, II

We have:

$$p_j = \frac{|c_j|}{\|\boldsymbol{c}\|_1} \implies \mathrm{Var}X = \|\boldsymbol{c}\|_1^2 - (\mathbf{1}^T \boldsymbol{c})^2.$$

In this case, if $\boldsymbol{c} = \boldsymbol{e}_j$, then $\mathrm{Var}X = 0$. (This was the "near" worst-case before.)

Importance sampling, II

We have:

$$p_j = \frac{|c_j|}{\|\boldsymbol{c}\|_1} \implies \mathrm{Var} X = \|\boldsymbol{c}\|_1^2 - (\mathbf{1}^T \boldsymbol{c})^2.$$

In this case, if $\boldsymbol{c} = \boldsymbol{e}_j$, then $\mathrm{Var} X = 0$. (This was the "near" worst-case before.)

This analysis can be lifted to the case when the inner product is $(m \times n)$-valued (i.e., a matrix). Like before, with $\boldsymbol{A} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k \times n}$,

$$\boldsymbol{AB} = \sum_{j \in [k]} \boldsymbol{a}_j \boldsymbol{b}_j^T.$$

With $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ the random matrix,

$$p_{\boldsymbol{X}}\left(\frac{1}{p_j} \boldsymbol{a}_j \boldsymbol{b}_j^T\right) = p_j \quad \implies \quad \mathbb{E} X = \boldsymbol{AB}.$$

Importance sampling, III

A direct computation yields that the expected Frobenius norm error is,

$$\mathbb{E}\left\|\boldsymbol{AB}-\frac{1}{N}\sum_{q\in[N]}\boldsymbol{X}_q\right\|_F^2=\frac{1}{N}\mathrm{trace}(\mathrm{Var}(\mathrm{vec}(\boldsymbol{X})))=\frac{1}{N}\sum_{(i,j)\in[m]\times[n]}\mathrm{Var}((AB)_{i,j})$$

$$=\frac{1}{N}\left(\sum_{j\in[k]}\frac{1}{p_j}\|\boldsymbol{a}_j\|_2^2\|\boldsymbol{b}_j\|_2^2-\|\boldsymbol{AB}\|_F^2\right).$$

This quadratic norm is miminized by choosing,

$$p_j=\frac{\|\boldsymbol{a}_j\|_2\|\boldsymbol{b}_j\|_2}{\sum_{q\in[k]}\|\|\boldsymbol{a}_q\|_2\|\boldsymbol{b}_q\|_2}\quad\Longrightarrow\quad\mathrm{trace}(\mathrm{Var}(\mathrm{vec}(\boldsymbol{X})))=\left(\sum_{q\in[k]}\|\boldsymbol{a}_q\|_2\|\boldsymbol{b}_q\|_2\right)^2-\|\boldsymbol{AB}\|_F^2.$$

This does give preasymptotic quantitative understanding of first- and second-moments of the estimator.

A note on practicality

We sort-of have a chicken vs egg problem: To compute $\|\boldsymbol{a}_q\|_2$ and $\|\boldsymbol{b}_q\|_2$ for all q naively, we require k-dependent complexity, which we're trying to avoid.

Sometimes a there is exploitable structure in matrices that allow us to compute these values.

Alteratively, if we can *approximate* these values, then we can still achieve similar results.

A note on practicality

We sort-of have a chicken vs egg problem: To compute $\|\boldsymbol{a}_q\|_2$ and $\|\boldsymbol{b}_q\|_2$ for all q naively, we require k-dependent complexity, which we're trying to avoid.

Sometimes a there is exploitable structure in matrices that allow us to compute these values.

Alteratively, if we can *approximate* these values, then we can still achieve similar results.

Namely, if we can choose the probabilities p_j so that for some $\tau \leqslant 1$,

$$p_j \geqslant \tau \frac{\|\boldsymbol{a}_j\|_2 \|\boldsymbol{b}_j\|_2}{\sum_{q \in [k]} \|\|\boldsymbol{a}_q\|_2 \|\boldsymbol{b}_q\|_2}$$

then the resulting quadratic expected error suffers a multiplicative $1/\tau$ penalty.

A note on practicality

We sort-of have a chicken vs egg problem: To compute $\|\boldsymbol{a}_q\|_2$ and $\|\boldsymbol{b}_q\|_2$ for all q naively, we require k-dependent complexity, which we're trying to avoid.

Sometimes a there is exploitable structure in matrices that allow us to compute these values.

Alteratively, if we can *approximate* these values, then we can still achieve similar results.

Namely, if we can choose the probabilities p_j so that for some $\tau \leqslant 1$,

$$p_j \geqslant \tau \frac{\|\boldsymbol{a}_j\|_2 \|\boldsymbol{b}_j\|_2}{\sum_{q\in[k]} \|\boldsymbol{a}_q\|_2 \|\boldsymbol{b}_q\|_2}$$

then the resulting quadratic expected error suffers a multiplicative $1/\tau$ penalty.

The point: we can sample *near*-optimally and get near-optimal results.

Moments to probabilities, I

We've computed the *expectation* of the error.

More practical information, such as the probability of failure, require more analysis. A simple, suboptimal strategy is to use e.g., *Markov's inequality*,

$$\Pr(X \geqslant t) \leqslant \frac{\mathbb{E}X}{t} \qquad (X \geqslant 0 \text{ wp1})$$

("Pf": $\mathbb{E}X = P(X \geq t)\,\mathbb{E}[X \mid X \geq t] + P(X < t)\,\mathbb{E}[X \mid X < t]$

$\geq P(X \geq t)\,\mathbb{E}[X \mid X \geq t] \geq t\,P(X \geq t)$)

Moments to probabilities, I

We've computed the *expectation* of the error.
More practical information, such as the probability of failure, require more analysis. A simple, suboptimal strategy is to use e.g., *Markov's inequality*,

$$\Pr(X \geqslant t) \leqslant \frac{\mathbb{E}X}{t} \qquad (X \geqslant 0 \text{ wp1})$$

To use this in our matrix multiplication setting, let,

$$Z = \left\| \boldsymbol{AB} - \frac{1}{N} \sum_{q \in [N]} \boldsymbol{X}_q \right\|_F^2, \qquad \mathbb{E}Z = \frac{1}{N} \left[\left(\sum_{q \in [k]} \|\boldsymbol{a}_q\|_2 \|\boldsymbol{b}_q\|_2 \right)^2 - \|\boldsymbol{AB}\|_F^2 \right] =: \frac{\beta}{N}.$$

Our goals are:

- Given $\epsilon > 0$, ensure that $Z \leqslant \epsilon\beta$.
- Given $\delta > 0$, ensure failure of the above with probability at most δ.

I.e., given ϵ, δ, when is it true that $\Pr(Z \geqslant \epsilon\beta) \leqslant \delta$?

$N \geq N(\epsilon,\delta)$ ⇑ ?

$\Pr(Z \geq \epsilon\beta) \leq \frac{\mathbb{E}Z}{\epsilon\beta} = \frac{\beta}{\epsilon\beta N} \leq \delta$

Moments to probabilities, II

We require $N \geqslant 1/(\delta\epsilon)$ for this to occur.

This is a precise sample complexity to achieve prescribed accuracy with prescribed error.

This means: if we choose $N \geqslant 1/(\delta\epsilon)$, then $\mathbb{E}Z \leqslant \epsilon\beta$ with probability at least $1 - \delta$.

(To achieve simplicity, we're kind of cheating here: this is a bound for *quadratic* error. Really we should worry about $\sqrt{Z}$. By using Jensen's inequality, $N \geqslant 1/(\delta\epsilon)^2$ is the sample requirement for ϵ-relative accuracy on $\sqrt{Z}$.)

Matrix multiplication summary

Using a simple concentration strategy, we have a random sampling algorithm (with probabilistic weights depending on the columw/row norms of $\boldsymbol{A}, \boldsymbol{B}$) that achieves a prescribed error with a prescribed probability.

- We can explicitly compute moments.
- A variance-like quadratic deviation can be minimized by choosing appropriate probabilities (that require knowledge of $\boldsymbol{A}, \boldsymbol{B}$).
- These moments can be transformed into failure probabilities through inequalities. (We used Markov's inequality.)
- This results in precise sample requirements to achieve (error, success).

Matrix multiplication summary

Using a simple concentration strategy, we have a random sampling algorithm (with probabilistic weights depending on the columw/row norms of $\boldsymbol{A}, \boldsymbol{B}$) that achieves a prescribed error with a prescribed probability.

- We can explicitly compute moments.
- A variance-like quadratic deviation can be minimized by choosing appropriate probabilities (that require knowledge of $\boldsymbol{A}, \boldsymbol{B}$).
- These moments can be transformed into failure probabilities through inequalities. (We used Markov's inequality.)
- This results in precise sample requirements to achieve (error, success).
- The resulting sampling complexity is not that great: ensuring an at-most 10% failure rate with 10% relative error requires 100 samples. (And this is to guarantee achieving the *quadratic* variance-type error.)
- We can do better...with some more work. The way we've transformed moments into probabilities is a very loose translation. Stronger, sharper results require more precise estimates of *concentration*.