2 Randomized Matrix Multiplication

1. Lecture
18.10.2022

Motivations:

- Basis transformation
- Greedy algorithm (later)
- Application to randomized SVD

Linear operations are the most basic function model → Problem of crucial importance.
How do we multiply matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$?

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & \dots \\ \dots & \dots \end{pmatrix}$$

Algorithm 1: Naive matrix multiplication

```
for i = 1 to m do
    for j = 1 to p do
        (AB)_ij = 0
        for k = 1 to n do
            (AB)_ij = (AB)_ij + A_ik B_kj
```

⇝ Computational cost of $O(m \cdot n \cdot p)$.

- *Can we do that faster?* Yes, Strassen's algorithm (based on tensor representations) and follow-ups give exact product in fewer operations.
  For $m = n = p$: best known order is $O(m^{2.37\dots})$
  → still large for m large
  → does not compete with standard approach due to implementation aspects.
- *Can we make it still faster if we are OK with approximate solutions?*
  → approximate solution ⇝ small *error*
  → randomized algorithm with small probability of failure.
- To do that, we need measure for size of error. → Norms on matrix space:

$$\|A\|_F = \sqrt{\operatorname{tr} A^*A} = \sqrt{\sum_{i,j} a_{ij}^2} \qquad \text{Frobenius norm}$$

$$\|A\| = \sup_{x:\|x\|=1} \|Ax\|_2 \qquad \text{Spectral norm}$$

The Frobenius norm will be used as a measure for error.

2. Lecture
25.10.2022

Observation:

$$AB = \sum_i \underbrace{A^{(i)}}_{i\text{-th column of } A} \cdot \underbrace{B_{(i)}}_{i\text{-th row of } B}$$

Goal: Approximate matrix product with fewer operations
Idea: Subsample and renormalize.

(i) **Subsampling**: sum only **random samples** of outer products, that is,

$$S_1 = \sum_{t=1}^{c} A^{(i_t)} B_{(i_t)}$$

where the i_t are drawn independently at random according to some probability measure ν on $[n] := \{1, ..., n\}$.

(ii) **Renormalization:** Calculate

$$\begin{aligned} \mathbb{E}S_1 &= c \cdot \mathbb{E}[A^{(i_1)} B_{(i_1)}] \\ &= c \cdot \sum_{j=1}^{n} \nu(j) A^{(j)} B_{(j)} \end{aligned}$$

We would like this to be AB. So we renormalize and consider

$$\boxed{S = \sum_{t=1}^{c} \frac{1}{c\nu(i_t)} A^{(i_t)} B_{(i_t)}} \implies \mathbb{E}S = AB \tag{2.1}$$

Algorithm 2: Randomized matrix multiplication

Input : $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}, c \in \mathbb{N}, p_1 = \nu(1), ..., p_n = \nu(n)$
Output: Approximate product $P \in \mathbb{R}^{m\times p}$

for $t = 1$ *to* c **do**
  Pick $i_t \in [n]$ i.i.d. according to ν
  Set $c^{(t)} = \frac{A^{(i_t)}}{\sqrt{cp_{i_t}}}$, $R_{(t)} = \frac{B_{(it)}}{\sqrt{cp_{i_t}}}$
end
return $P = CR$ *(calculated using your favorite deterministic algorithm)*

- How should we choose the measure ν?

  → Answer potentially depends on the error measure.

  Here: difference in Frobenius norm $\|M\|_F^2 = \sum_{i,j} M_{ij}^2$.

- First step: Calculate the variance of the approximate products.

Lemma 2.1 (Variance of the approximate products)**.** Given matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, define S as in (2.1). Then

$$\operatorname{Var} S_{ij} = \frac{1}{c} \sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{\nu(k)} - \frac{1}{c}(AB)_{ij}^2. \tag{2.2}$$

Proof. Fix i, j and set, for $t \in [c]$,

$$X_t := \left(\frac{A^{(i_t)} B_{(i_t)}}{c\nu(i_t)} \right)_{ij} = \frac{A_{ii_t} B_{i_t j}}{c\nu(i_t)}.$$

Thus

$$\mathbb{E}X_t^2 = \sum_{k=1}^{n} \nu(k) \left(\frac{A_{ik}B_{kj}}{c\nu(k)} \right)^2 = \frac{1}{c^2} \sum_{k=1}^{n} \frac{1}{\nu(k)} A_{ik}^2 B_{kj}^2,$$

and similarly

$$\mathbb{E}X_t = \sum_{k=1}^{n} \nu(k) \frac{A_{ik}B_{kj}}{c\nu(k)} = \frac{1}{c}(AB)_{ij}.$$

Note that this is consistent with $\mathbb{E}S = AB$. Moreover,

$$\begin{aligned}
\mathbb{E}S_{ij}^2 = \mathbb{E}\left(\sum_{t=1}^{c} X_t \right)^2 &\overset{\text{indep.}}{=} \sum_{t=1}^{c} \mathbb{E}X_t^2 + \sum_{t_1,t_2,t_1 \neq t_2}^{c} \mathbb{E}X_{t_1}\mathbb{E}X_{t_2} \\
&= \frac{1}{c} \sum_{k=1}^{n} \frac{1}{\nu(k)} A_{ik}^2 B_{kj}^2 + \underbrace{c(c-1)}_{\text{summands}} \left(\frac{1}{c}(AB)_{ij} \right)^2 \\
&= \frac{1}{c} \sum_{k=1}^{n} \frac{1}{\nu(k)} A_{ik}^2 B_{kj}^2 + \frac{c-1}{c}(AB)_{ij}^2.
\end{aligned}$$

Consequently,

$$\operatorname{Var} S_{ij} = \mathbb{E}S_{ij}^2 - \underbrace{(\mathbb{E}S_{ij})^2}_{(AB)_{ij}^2} = \frac{1}{c} \sum_{k=1}^{n} \frac{1}{\nu(k)} A_{ik}^2 B_{kj}^2 - \frac{1}{c}(AB)_{ij}^2,$$

which proves the lemma. ■

Lemma 2.2 (Expected error of approximate products)**.** Given matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, define the approximate product S as in (2.1), then

$$\mathbb{E}[\|AB - S\|_F^2] = \sum_{k=1}^{n} \frac{\|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2}{c\nu(k)} - \frac{1}{c}\|AB\|_F^2. \tag{2.3}$$

Proof. Note that

$$\begin{aligned}
\mathbb{E}[\|AB - S\|_F^2] &= \sum_{i=1}^{m} \sum_{j=1}^{p} \mathbb{E}[(AB - S)_{ij}^2] \\
&= \sum_{i=1}^{m} \sum_{j=1}^{p} \underbrace{\mathbb{E}[(S - \mathbb{E}S)_{ij}^2]}_{\operatorname{Var} S_{ij}}.
\end{aligned}$$

Thus from Lemma 2.1, it follows that

$$\begin{aligned}
\mathbb{E}[\|AB - S\|_F^2] &= \frac{1}{c} \sum_{k=1}^{n} \frac{1}{\nu(k)} \left(\sum_{i=1}^{m} A_{ik}^2 \right) \left(\sum_{j=1}^{p} B_{kj}^2 \right) - \frac{1}{c} \sum_{i,j} (AB)_{ij}^2 \\
&= \sum_{k=1}^{n} \frac{\|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2}{c\nu(k)} - \frac{1}{c}\|AB\|_F^2
\end{aligned}$$

which completes the proof. ■

Best we can hope for: the error $\|AB - S\|_F^2$ is consistently close to expectation ("concentration phenomenon"). Thus we should choose the measure ν such that

$$\sum_{k=1}^{n} \frac{\|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2}{c\nu(k)}$$

is small. (Note that the second summand is independent of measure v.)
Let p denote the vector of mass distribution of ν, i.e., $p_k = \nu(k)$. Then we are looking for the minimizer of the smooth function

$$f(p) = \sum_{k=1}^{n} \frac{\|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2}{p_k}$$

over

$$P := \left\{ p \in [0,1] : \underbrace{\sum_{k=1}^{n} p_k}_{=:g(p)} = 1 \right\}.$$

Lagrange multiplier formulation: Extremum must satisfy, for some $\lambda \in \mathbb{R}$

$$\begin{aligned} 0 &= \nabla f(p) + \lambda \nabla g(p) \\ &= \left(-\frac{\|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2}{p_k^2} \right)_{k=1}^{n} + \lambda (\underbrace{1, ..., 1}_{n \text{ times}}) \end{aligned}$$

This implies

$$\lambda p_k^2 = \|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2.$$

Thus,

$$\sqrt{\lambda} \overset{\sum_k p_k = 1}{=} \sum_{k=1}^{n} \sqrt{\lambda} p_k = \sum_{k=1}^{n} \|A^{(k)}\|_2 \|B_{(k)}\|_2.$$

If that sum vanishes, the product AB vanishes. If not, we obtain the unique critical point

$$\boxed{p = \left(\frac{\|A^{(j)}\|_2 \|B_{(j)}\|_2}{\sum_{k=1}^{n} \|A^{(k)}\|_2 \|B_{(k)}\|_2} \right)_{j=1}^{n}}. \tag{2.4}$$

If $p_k \to 0$ for some k with $\|A^{(k)}\| \neq 0 \neq \|B_{(k)}\|_2$, then $f(p) \to \infty$, so the critical point must be the minimizer.
Choosing $\nu(j) = p_j$, we obtain

$$\boxed{\mathbb{E}[\|AB - S\|_F^2] = \frac{1}{c} \left(\sum_{k=1}^{n} \|A^{(k)}\|_2 \|B_{(k)}\|_2 \right)^2 - \frac{1}{c} \|AB\|_F^2}. \tag{2.5}$$

Remark. For rank 1 matrices, it holds that

$$\boxed{\|A^{(k)}\|_2 \|B_{(k)}\|_2 = \|A^{(k)} B_{(k)}\| = \|A^{(k)} B_{(k)}\|_F}. \tag{2.6}$$

The interpretation of optimal sampling strategy: we bias the random samples towards rank one components which are larger in norm.

In our **implementation**, we have two different objectives:

- minimize number of mathematical operations
- minimize storage space requirements

In addition to carrying out the reduced size multiplication [$O(m \cdot c \cdot p)$ operations], we need to compute (2.4) (i.e., the probability measure ν). We need

- $O(m)$ operations to compute each $\|A^{(n)}\|_2$,
- $O(p)$ operations to compute each $\|B_{(n)}\|_2$,

and in total $O(mn + np)$.

Space requirement: $O(n)$ to store all of them.

How do we then *sample*? We need a model for accessing data. *Model*: Data streaming, i.e., data "stream by", one computes on the stream without storing. → "pass efficiency".

---

3. Lecture
08.11.2022

Goal: Approximate AB by

$$\sum_{k=1}^{c} \frac{1}{c\nu(i_k)} A^{(i_k)} B_{(i_k)} =: S,$$

where i_k are i.i.d. drawn from probability measure μ.

Optimal choice:

$$\nu(k) = \frac{\|A^{(k)}\|_2 \cdot \|B_{(k)}\|_2}{\sum_{j=1}^{n} \|A^{(j)}\|_2 \|B_{(j)}\|_2}.$$

Next: Model for accessing data.

Definition 2.3 (Pass efficient model)**.** In the pass efficient model, the only access an algorithm has to the data is via a pass, i.e., a sequential read of the entire data set. An algorithm is *pass-efficient*, if it requires a small constant number of passes and additional space and time, which are *sub-linear* in the length of the data stream.

- idealized model
- *Motivation:* In many applications, one has the ability to store or generate larger amounts of data, but has random access to only linked amounts.

g is sublinear in $N \iff g(N) = o(N) \iff \forall C > 0 \, \exists N_0 \in \mathbb{N} \, \forall N \geqslant N_0 : g(N) \leqslant CN.$

Remark. As we "do something" whenever we read an entry, the number of operations is in fact linear, main requirement: sublinear access queries.

Algorithm 3: Select algorithm

Input : $\{a_1, ..., a_n\}, a_i \geqslant 0, \sum_i a_i > 0$
Output: i^*, a_{i^*}

$D = 0$ ← normalization factor
for $i = 1$ *to* n **do**
  $D = D + a_i$
  with probability $\frac{a_i}{D}$ let $i^* = i, a_{i^*} = a_i$ (or w.p. 1 if $a_i = D = 0$)
end
return i^*, a_{i^*}

Remark. Indeed, this algorithm returns each i with probability $\frac{a_i}{\sum_j a_j}$.

Example. Let $a_1 = a_2 = a_3 = 5$. Then,

- $i = 1$: $D = 5$ with $P = \frac{5}{5} = 1$. Choose $i^* = 1$.
- $i = 2$: $D = 10$ with $P = \frac{5}{10} = \frac{1}{2}$. Then $P(i^* = 1) = P(i^* = 2) = \frac{1}{2}$. Choose $i^* = 2$.
- $i = 3$: $D = 15$ with $P = \frac{5}{10} = \frac{1}{3}$. Then by independence, $P(i^* = 1) = P(i^* = 2) == P(i^* = 3) = \frac{1}{3}$. Choose $i^* = 3$.

Lemma 2.4. Suppose that the selection algorithm is applied with inputs $\{a_1, ..., a_n\}$, $a_i \geqslant 0$. Then the additional storage space required is $O(1)$ and the output i^* satisfies $P(i^* = i) = \frac{a_i}{\sum_{j=1}^n a_j}$.

Proof. Note that only the current values of i^*, a_{i^*}, and D must be retained, which corresponds to $O(1)$ space.
The reminder of the proof is by induction. After reading a_1, one has $i^* = 1$ w.p. $\frac{a_1}{a_1} = 1$ (provided $a_1 = 0$, otherwise 1 by definition). As the induction hypothesis, assume that after reading $a_1, ..., a_l$, the variable i^* satisfies

$$P(i^* = i) = \frac{a_i}{D_l} \quad \text{for } i \in [l],$$

where $D_l = \sum_{i=1}^l a_i$. Upon reading a_{l+1}, the algorithm sets $i^* = l + 1$ w.p. $\frac{a_{l+1}}{D_{l+1}}$ and retains its previous value w.p. $1 - \frac{a_{l+1}}{D_{l+1}} = \frac{D_l}{D_{l+1}}$. Thus by independence, after reading $a_1, ..., a_{l+1}$, one has for $i < l + 1$:

$$P(i^* = i) = \frac{a_i}{D_l} \cdot \frac{D_l}{D_{l+1}} = \frac{a_i}{D_{l+1}}.$$

and for $i = l + 1$, the same holds by construction. This completes the induction step and the results follows as $\frac{a_i}{D_n} = \frac{a_i}{\sum_{j=1}^n a_j}$. ■

To draw c independent samples, the select algorithm is repeated c times. Thus the total number of operations required to select c indices independently at random is $O(c \cdot n)$, where c is the number of passes and n is the number of elements/operations per pass. ⇝ Total number of operations $O(mn + np + cn + mcp)$.

Observation: The only cubic term in the above expression is mcp. Thus, c determines the number of operations. To determine the size of c, we, however, face the following conflicting goals:

- Number of summands c should be as small as possible.
- Error "in most cases" should be as small as possible.
- Probability of exceptional cases ("failure") should be as small as possible.

Those goals can be reformulated as archiving

$$\|AB - S\|_F \leqslant \varepsilon \|A\|_F \|B\|_F \quad \text{w.p. } \geqslant 1 - \delta \tag{2.7}$$

for ε, δ and c as small as possible.

Remark. Assuming that $S = CR \in \mathbb{R}^{m \times p}$ is the output of Algorithm 2 for the parameter c with

$$c \geqslant \frac{1}{\delta^2 \varepsilon^2}$$

and optimal probability defined in (2.4), then it holds

$$\|AB - S\|_F \leqslant \varepsilon \|A\|_F \|B\|_F \quad \text{w.p. } \geqslant 1 - \delta \tag{2.8}$$

for arbitrary $\varepsilon, \delta > 0$.

Proof. (Exercise 3.2) First, Markov inequality implies

$$P(\|AB - S\|_F \geqslant \varepsilon \|A\|_F \|B\|_F) \leqslant \frac{\mathbb{E}[\|AB - S\|_F]}{\varepsilon \|A\|_F \|B\|_F}.$$

We now estimate the numerator $\mathbb{E}[\|AB - S\|_F]$

$$\begin{aligned}
\mathbb{E}[\|AB - S\|_F^2] &\overset{(2.5)}{=}_{\text{opt. prob.}} \frac{1}{c} \underbrace{\left(\sum_{k=1}^{n} \|A^{(k)}\|_2 \|B_{(k)}\|_2 \right)^2}_{\overset{\text{C.S.}}{\leqslant} (\sum_{k=1}^{n} \|A^{(k)}\|_2^2)(\sum_{k=1}^{n} \|B_{(k)}\|_2^2)} - \frac{1}{c}\|AB\|_F^2 \\
&\leqslant \frac{1}{c} \left(\sum_{k=1}^{n} \|A^{(k)}\|_2^2 \right) \left(\sum_{k=1}^{n} \|B_{(k)}\|_2^2 \right) \\
&= \frac{1}{c} \|A\|_F^2 \|B\|_F^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
P(\|AB - S\|_F \geqslant \varepsilon \|A\|_F \|B\|_F) &\leqslant \frac{\mathbb{E}[\|AB - S\|_F]}{\varepsilon \|A\|_F \|B\|_F} \\
&\leqslant \frac{1}{\sqrt{c}\varepsilon} \overset{c \geqslant \frac{1}{\delta^2 \varepsilon^2}}{\leqslant} \delta,
\end{aligned}$$

and thus $P(\|AB - S\|_F \leqslant \varepsilon \|A\|_F \|B\|_F) \geqslant 1 - \delta$. ∎

This result may suffice if 10% failures are fine. But for very small probabilities of failure, we need very large number of measurements [see Exercise 3.2 (c)]. But why can we hope for something better?

- S is average of independent random matrices.
- Law of large numbers suggests convergence to the mean, which is AB.

However, we still face two issues:

(i) We need non-asymptotic behavior rather than limit as $n \to \infty$ (as in LLN).

(ii) We need matrix version in Frobenius norm.

Idea: Do not look at this as a matrix problem, rather consider the real-valued function

$$F : (i_1, ..., i_c) \mapsto \underbrace{\left\| \sum_{t=1}^{c} \frac{1}{c\nu(i_t)} A^{(i_t)} B_{(i_t)} - \mathbb{E}\left[\sum_{t=1}^{c} \frac{1}{c\nu(i_t)} A^{(i_t)} B_{(i_t)} \right] \right\|_F}_{\|S - AB\|_F}. \tag{2.9}$$

Observations:

- Inputs are i.i.d. random variables with values in $[n]$.
- $\mathbb{E}[F]$ is under control (via Lemma 2.2).

Tool: Version of Mcdiarmid's inequality.

4. Lecture 22.11.2022

Theorem 2.5 (Mcdiarmid's inequality)**.** Let $S \subset \mathbb{N}$ and let $X_1, ..., X_n$ be independent random variables with values in S. Let $F : S^n \to \mathbb{R}$ be a map fulfilling the following *bounded differences property*: There exists $\Delta > 0$ such that for all $x_1, ..., x_n, x_1', ..., x_n'$, it holds that

$$|F(x_1, ..., x_n) - F(x_1', ..., x_n')| \leqslant \Delta \sum_{i=1}^{n} \mathbb{1}_{\{x_i \neq x_i'\}}. \tag{2.10}$$

Then for any $t > 0$,

$$P(F(X_1, ..., X_n) - \mathbb{E}[F(X_1, ..., X_n] \geqslant t) \leqslant \exp\left(\frac{-2t^2}{n\Delta^2}. \right) \tag{2.11}$$

Proof. See Exercise 3.3. ■

With this tool, we can show the following bound with a much better scaling in δ.

Theorem 2.6. Suppose $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}, c \in [n]$, and ν is a probability measure on $[n]$ such that, for some constant $0 < \beta \leqslant 1$

$$\nu(k) \geqslant \frac{\beta \|A^{(k)}\|_2 \|B_{(k)}\|_2}{\sum_{k'=1}^{n} \|A^{(k')}\|_2 \|B_{(k')}\|_2}. \tag{2.12}$$

We construct S using Algorithm 2 above. Then for all $\delta \in (0, 1)$ and the associated $\eta = 1 + \sqrt{\frac{2}{\beta} \log(\frac{1}{\delta})}$ it holds that

$$\|AB - S\|_F \leqslant \frac{\eta}{\sqrt{\beta c}} \|A\|_F \|B\|_F \quad \text{w.p.} \quad \geqslant 1 - \delta. \tag{2.13}$$

The role of β is to measure the amount of deviation from the optimal sampling density.

Proof. First note that

$$\begin{aligned}
\mathbb{E}\|AB-S\|_F &\overset{\text{Jensen}}{\leqslant} \sqrt{\mathbb{E}\|AB-S\|_F^2} \\
&\overset{(2.3)}{=} \sqrt{\sum_{k=1}^n \frac{\|A^{(k)}\|_2^2\|B_{(k)}\|_2^2}{c\nu(k)} - \frac{1}{c}\underbrace{\|AB\|_F^2}_{\geqslant 0}} \\
&\overset{(2.12)}{\leqslant} \frac{1}{\sqrt{\beta c}} \underbrace{\sum_{k=1}^n \|A^{(k)}\|_2\|B_{(k)}\|_2}_{=\langle(\|A^{(k)}\|_2)_{k=1}^n,(\|B_{(k)}\|_2)_{k=1}^n\rangle} \\
&\overset{\text{C.S.}}{\leqslant} \frac{1}{\sqrt{\beta c}} \underbrace{\left(\sum_{k=1}^n \|A^{(k)}\|_2^2\right)^{\frac{1}{2}}}_{\|A\|_F} \underbrace{\left(\sum_{k=1}^n \|B_{(k)}\|_2^2\right)^{\frac{1}{2}}}_{\|B\|_F} = \frac{1}{\sqrt{\beta c}}\|A\|_F\|B\|_F.
\end{aligned} \tag{2.14}$$

Thus it remains to show that

$$\|AB-S\|_F - \mathbb{E}[\|AB-S\|_F] \leqslant \frac{\eta-1}{\sqrt{\beta c}}\|A\|_F\|B\|_F \quad \text{w.p.} \quad \geqslant 1-\delta.$$

To prove this, we apply Theorem 2.5 with (2.9):

$$F : (i_1,...,i_c) \mapsto \left\| \sum_{t=1}^c \frac{1}{c\nu(i_t)} A^{(i_t)}B_{(i_t)} - AB \right\|_F,$$

and $\hat{n} = c$,[1] $S = [n]$. We first need to check the bounded difference property (2.10). We fix $r \in [c]$. Then

$$\begin{aligned}
&|F(i_1,...,i_r,i_{r+1},...,i_c) - F(i_1,...,i_r',i_{r+1},...,i_c)| \\
&= \left| \left\| \frac{1}{c}\sum_{t=1}^c \frac{1}{\nu(i_t)} A^{(i_t)}B_{(i_t)} - AB \right\|_F - \left\| \frac{1}{c}\sum_{t=1,t\neq r}^c \frac{1}{\nu(i_t)} A^{(i_t)}B_{(i_t)} - AB + \frac{1}{c}\frac{1}{\nu(i_r')}A^{(i_r')}B_{(i_r')} \right\|_F \right| \\
&\overset{\text{rev. }\Delta\text{-ineq}}{\leqslant} \left\| \frac{1}{c\nu(i_r)}A^{(i_r)}B_{(i_r)} - \frac{1}{c\nu(i_r')}A^{(i_r')}B_{(i_r')} \right\|_F \\
&\overset{\Delta\text{-ineq}}{\underset{(2.6)}{\leqslant}} \frac{1}{c\nu(i_r)}\|A^{(i_r)}\|_2\|B_{(i_r)}\|_2 + \frac{1}{c\nu(i_r')}\|A^{(i_r')}\|_2\|B_{(i_r')}\|_2 \\
&\leqslant \frac{2}{c}\max_{\alpha\in[n]}\frac{\|A^{(\alpha)}\|_2\|B_{(\alpha)}\|_2}{\nu(\alpha)} \\
&\overset{(2.12)}{\leqslant} \frac{2}{c\beta}\sum_{k'=1}^n \|A^{(k')}\|_2\|B_{(k')}\|_2 \overset{\text{C.S.}}{\leqslant} \underbrace{\frac{2}{\beta c}\|A\|_F\|B\|_F}_{=:\Delta}.
\end{aligned}$$

By the triangle inequality,this implies the bounded difference inequalities for arguments that differ in more than one entry and the same Δ. The result follows from Theorem 2.5 noting that $\exp(-\frac{\beta}{2}(\eta-1)^2) = \delta$. ■

[1]Here $\hat{n}$ refers to the n from Theorem 2.5.

3 Randomized Principal Component Analysis

Motivation. Many high-dimensional data sets are intrinsically low-dimensional, i.e., they can be well approximated by a lower dimensional subspace. For a given matrix $A \in \mathbb{R}^{m\times n}$, we want to find a lower dimensional embedding $\tilde{A}_k$ of rank k close to A (cf. principal component analysis).

Definition 3.1 (Best k-rank approximation)**.** Given $A \in \mathbb{R}^{m\times n}$, $A_k \in \mathbb{R}^{m\times n}$ is a *best rank-k approximation to* A with respect to a norm $|||\cdot|||$ if $A_k \in \underset{B\in\mathbb{R}^{m\times n},\operatorname{rk} B\leqslant k}{\operatorname{argmin}} |||A-B|||$.

Remark. With respect to the spectral norm and the Frobenius norm, A_k can be computed via the singular value decomposition. If the SVD of A is

$$U \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{pmatrix} V^*,$$

then the matrix

$$A_k = U \begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_k & \\ 0 & & & 0 \end{pmatrix} V^*$$

is the best rank-k approximation to A.

However, to compute the full SVD of A, we have a complexity of $O(\min(m^2n, mn^2))$, i.e. is cubic. Note that to archive our goal (finding $\tilde{A}_k$ of rank k such that $A-\tilde{A}_k$ and $A-A_k$ are comparable in size (up to constant)), a *crucial step* is to find left singular vectors $u_1, ..., u_k$, i.e. matrix U_k, corresponding to $\sigma_1, ..., \sigma_k$. To reduce complexity, we could first *project* A, then compute SVD of the smaller matrix, and infer approximation of U_k. For *projection*, we will use *randomized Hadamard transform*.

Definition 3.2 (Hadamard transform)**.** Let $n \in \mathbb{N}$ be a power of 2. Then the $n \times n$ **Hadamard matrix** $\widetilde{H}_n$ is defined recursively as follows

$$\widetilde{H}_n = \begin{pmatrix} \widetilde{H}_{\frac{n}{2}} & \widetilde{H}_{\frac{n}{2}} \\ \widetilde{H}_{\frac{n}{2}} & -\widetilde{H}_{\frac{n}{2}} \end{pmatrix} \quad \text{with} \quad \widetilde{H}_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}. \tag{3.1}$$

The **normalized Hadamard transform** is

$$H = H_n = \frac{1}{\sqrt{n}}\widetilde{H}_n.$$

Furthermore, for a random diagonal matrix $D \in \mathbb{R}^{n\times n}$ with independent entries D_{ii} with $P(D_{ii} = 1) = P(D_{ii} = -1) = \frac{1}{2}$, HD is called the **randomized Hadamard transform**.

Remark. The Hadamard transform has following properties:

(i) The randomized Hadamard transform HD is orthogonal, because the normalized Hadamard transform H and the Rademacher matrix D are orthogonal [Ex. 5.1].

(ii) Computing Hx (and hence HDx for x fixed needs $O(n \log n)$ operations (similar to fast Fourier transform, Hadamard $\hat{=}$ Fourier transform in $\mathbb{Z}_2$).

Intuition for Hadamard transform: HD "spreads out" mass/energy. We hope that after applying HD, it suffices to subsample according to uniform distribution (hence independent of A). This is made precise by the following lemma:

5. Lecture 29.11.2022

Lemma 3.3. Let $U \in \mathbb{R}^{n \times d}$ be a matrix with orthogonal columns and let the product HD be the $n \times n$ randomized Hadamard transform as introduced in Definition 3.2. Then, it holds that

$$\|(HDU)_{(i)}\|_2^2 \leqslant \frac{2d \log(40nd)}{n} \quad \forall i = 1, ..., n \quad \text{w.p.} \ \geqslant 1 - \frac{1}{20}.$$

Proof. Consider $(HDU)_{ij}$ for some $(i, j) \in [n] \times [d]$. As D is diagonal, one has

$$(HDU)_{ij} = \sum_{l=1}^{n} H_{il} D_{ll} U_{lj} = \sum_{l=1}^{n} D_{ll}(H_{il} U_{lj}).$$

As D_{ll} are i.i.d. random signs, $|D_{ll}(H_{il}U_{lj})| \leqslant |(H_{il}U_{lj})|$ a.s., and $\mathbb{E}[D_{ll}(H_{il}U_{lj})] = 0$, Hoeffding's inequality implies for all $t > 0$

$$\begin{aligned} P\left(\left|\sum_{l=1}^{n} D_{ll}(H_{il}U_{lj})\right| \geqslant t\right) &\leqslant 2\exp\left(-\frac{t^2}{2\sum_{l=1}^{n}(H_{il}U_{lj})^2}\right) \\ &\overset{(*)}{=} 2\exp\left(-\frac{nt^2}{2}\right), \end{aligned}$$

where $(*)$ comes from the fact

$$\sum_{l=1}^{n}(H_{il}U_{lj})^2 \overset{|H_{il}|=\frac{1}{\sqrt{n}}}{=} \frac{1}{n}\sum_{l=1}^{n} U_{lj}^2 = \frac{1}{n}\|U^{(j)}\|_2^2 \overset{U \text{ orth.}}{=} \frac{1}{n}.$$

Consider

$$\delta = 2\exp\left(-\frac{nt^2}{2}\right)$$

and choose $t := \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{n}}$, we get

$$P\left(|(HDU)_{ij}| \geqslant \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{n}}\right) = P\left(\left|\sum_{l=1}^{n} D_{ll}(H_{il}U_{lj})\right| \geqslant \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{n}}\right) \leqslant \delta. \tag{3.2}$$

Choosing $\delta = \frac{1}{20nd}$, we obtain

$$\begin{aligned} & P\left(\exists i \in [n] : \|(HDU)_{(i)}\|_2^2 \geqslant \frac{2d\log(40nd)}{n}\right) \\ \overset{(*)}{\leqslant}\ & P\left(\exists (i,j) : |(HDU)_{ij}| \geqslant \sqrt{\frac{2\log(40nd)}{n}}\right) \\ \overset{\text{Subadd.}}{\leqslant}\ & \sum_{i=1}^{n}\sum_{j=1}^{d} P\left(|(HDU)_{(ij)}| \geqslant \sqrt{\frac{2\log(40nd)}{n}}\right) \\ \overset{\substack{(3.2)\\ \frac{2}{\delta}=40nd}}{\leqslant}\ & \frac{1}{20nd} \leqslant \frac{1}{20}, \end{aligned}$$

where $(*)$ comes from the fact

$$\left(\forall\,(i,j) : |(HDU)_{ij}| \leqslant \sqrt{\frac{2\log(40nd)}{n}}\right) \implies \left(\forall\, i \in [n] : \|(HDU)_{(i)}\|_2^2 \leqslant \frac{2d\log(40nd)}{n}\right).$$

Considering the complement yields claim. ■

Remark. The result holds even for arbitrary $U \in \mathbb{R}^{n\times D}$ (not necessarily orthogonal), see Exercise 5.2.

Idea: Apply randomized Hadamard transformation from one side, e.g., postmultiplying $A \in \mathbb{R}^{m\times n}$ by $(HD)^T$, forming $ADH \in \mathbb{R}^{m\times n}$. Then sample (uniformly at random) c columns from ADH, thus forming a smaller matrix $C \in \mathbb{R}^{m\times c}$. From C, construct approximations of the top k singular values of A.

Algorithm 4: Randomized PCA

Input : $A \in \mathbb{R}^{m\times n}$, rank parameter $k \ll \min\{m,n\}$, error parameter $\varepsilon \in \left(0, \frac{1}{2}\right)$, name of samples $c \in \mathbb{N}$.

Output: $\widetilde{U}_k \in \mathbb{R}^{m\times k}$

set $S = 0^{n\times c}$ (all entries are 0) *//sample matrix*

for $t = 1$ *to* c **do**

  // i.i.d. trials with replacement

  select uniformly at random $i_t \in [n]$ and set $S^{(t)} = \sqrt{\frac{n}{c}} e_{i_t}$

end

compute $C = ADHS$ where HD is the randomized Hadamard transform

compute U_C, an ONB for the column space of C (e.g., via SVD)

compute $W = U_C^T A$ and compute its top k singular vectors $U_{W,k}$ (or less if $\operatorname{rank} W < k$)

return $\widetilde{U}_k = U_C U_{W,k} \in \mathbb{R}^{m\times k}$

Theorem 3.4. Let $A \in \mathbb{R}^{m\times n}$, let k be a rank parameter, and let $\varepsilon \in (0, \frac{1}{2})$. If we set

$$c \geqslant c_0 \frac{k \log n}{\varepsilon^2} \left(\log \frac{k}{\varepsilon^2} + \log\log n \right)$$

for a fixed constant c_0, then with probability $\geqslant 0.85$, Algorithm 4 returns a matrix $\widetilde{U}_k \in \mathbb{R}^{m\times k}$ such that

$$\|A - \widetilde{U}_k \widetilde{U}_k^T A\|_F \leqslant (1+\varepsilon)\|A - A_k\|_F.$$

The running time of the algorithm is $O(mnc)$. (Recall: A_k denotes the best k-rank approximation of A.)

Remark 3.5. Repeating Algorithm 4 $\lceil \log \frac{1}{\delta} / \log C(\delta) \rceil$-times and keeping the matrix $\widetilde{U}_k$ that minimizes $\|A - \widetilde{U}_k \widetilde{U}_k^T A\|_F$ reduces the failure probability to at most δ for $\delta \in (0,1)$.

Proof strategy of Theorem 3.4. We split $A = A_k + A_{k,\perp}$ for A_k the best rank-k approximation. We first show

$$\|A - \widetilde{U}_k \widetilde{U}_k^T A\|_F^2 \leqslant \|A_k - U_C U_C^T A_k\|_F^2 + \|A_{k,\perp}\|_F^2 \tag{3.3}$$

So in the reminder, we "only" need to control the first term on the right hand side. (The second yields $\|A - A_k\|_F$ on the right hand side of the statement we aim to prove. In order to show this inequality, we need the following lemma.

Lemma 3.6. Let U_C and $\widetilde{U}_k$ be as in Algorithm 4, i.e. the columns of U_C are an ONB of range(C). Then

$$A - \widetilde{U}_k \widetilde{U}_k^T A = A - U_C (U_C^T A)_k \tag{3.4}$$

In addition, $U_C(U_C^T A)_k$ is the best rank-k approximation of A with respect to $\|\cdot\|_F$ that lies within range(C), i.e.,

$$\|A - U_C(U_C^T A)_k\|_F^2 = \min_{\operatorname{rank}(Y)\leqslant k} \|A - U_C Y\|_F^2 \tag{3.5}$$

Proof. Recall that $\widetilde{U}_k = U_C U_{W,k}$, where $U_{W,k}$ is the matrix of the top k left singular vectors of $W = U_C^T A$. In other words, if $W = U_W \Sigma_W V_W^T$ denotes the SVD of W, the best k-term approximation W_k of W with respect to $\|\cdot\|_F$ has a singular value decomposition

$$W_k = U_{W,k} \Sigma_{W,k} V_W, \tag{3.6}$$

where $\Sigma_{W,k}$ consists of the first k rows of Σ_W. Consequently,

$$\begin{aligned} A - \widetilde{U}_k \widetilde{U}_k^T A =& A - U_C U_{W,k} U_{W,k}^T U_C^T A \\ =& A - U_C U_{W,k} \underbrace{\underbrace{U_{W,k}^T U_W}_{(\mathrm{id}_k\ \mathbf{0})} \Sigma_W}_{\Sigma_{W,k}} V_W \\ =& A - U_C W_k \\ =& A - U_C (U_C^T A)_k. \end{aligned} \tag{3.7}$$

To prove (3.5), we will use a homework result showing that

$$\|A - U_C Y\|_F^2 = \|(I - U_C U_C^T)A\|_F^2 + \|U_C^T A - Y\|_F^2 \quad \forall Y \text{ with } \operatorname{rk}(Y) \leqslant k. \tag{3.8}$$

Now the minimum on the right hand side (and hence also on the left hand side) under the constraint rank $\leqslant k$ is archived at $y = (U_C^T A)_k$, which shows (3.5). ■

6. Lecture
06.12.2022