CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru

University of Texas, Austin

Spring 2024

Lecture 6: Approximate matrix product and sampling

Outline

Why randomization?

- *Modern data applications*: massive data, computationally expensive problems.
- *Approximate solutions* suffice in many situations.
- **Randomized sampling and sketching** allow us to design approximation algorithms with provable error guarantees.
- Probabilistic error bounds. E.g., the (ϵ, δ) type bounds.

Product and norms using randomization

If a random distribution on $\boldsymbol{s} \in \mathbb{R}^n$ has entries s_i with:

- $\mathbb{E}[\mathrm{s}_i^2] = 1$ for $i = [n]$ and $\mathbb{E}[\mathrm{s}_i \mathrm{s}_j] = 0$ for $i, j = [n], i \neq j$.
- Then, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we have

$$\mathbb{E}[\langle \boldsymbol{s} \cdot \boldsymbol{x}, \boldsymbol{s} \cdot \boldsymbol{y} \rangle] = \mathbb{E}[(\boldsymbol{s}^\top \boldsymbol{x}) \cdot (\boldsymbol{s}^\top \boldsymbol{y})] = \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{y}] = \boldsymbol{x}^\top \boldsymbol{y}$$

- In particular, $\mathbb{E}[(\boldsymbol{s}^\top \boldsymbol{y})^2] = \mathbb{E}[\boldsymbol{y}^\top \boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{y}] = \boldsymbol{y}^\top \boldsymbol{y} = \|\boldsymbol{y}\|^2$.

$$\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^\top] = \begin{bmatrix} \mathrm{s}_1^2 & \mathrm{s}_1 \mathrm{s}_2 & \cdots & \mathrm{s}_1 \mathrm{s}_n \\ \mathrm{s}_2, \mathrm{s}_1 & \mathrm{s}_2^2 & & \vdots \\ \vdots & & \ddots & \\ \mathrm{s}_n, \mathrm{s}_1 & \cdots & & \mathrm{s}_n^2 \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 \end{bmatrix}$$

Sketching and Sampling

Sketching:

- Suppose $\mathrm{s}_i \sim \mathcal{N}(0,1)$ and independent.
- We have $\mathbb{E}[\mathrm{s}_i] = 0$, $\mathbb{E}[\mathrm{s}_i^2] = \mathrm{Var}(\mathrm{s}_i) = 1$.
- For $i \neq j$, independence implies $\mathbb{E}[\mathrm{s}_i \mathrm{s}_j] = \mathbb{E}[\mathrm{s}_i]\mathbb{E}[\mathrm{s}_i] = 0$.

Sampling:

- Suppose we pick $i \in [n]$ uniformly with probability $\frac{1}{n}$ and set $\mathrm{s}_i \leftarrow \sqrt{n}, 0$ o.w.
- We have $\mathbb{E}[\mathrm{s}_i^2] = \frac{1}{n}\sqrt{n}^2 + (1 - \frac{1}{n})0 = 1$.
- For $i \neq j$ if $\mathrm{s}_i \neq 0 \implies \mathrm{s}_j = 0$, so $\mathrm{s}_i \mathrm{s}_j = 0$.

Randomized techniques

With repetitions and better distributions, randomization can be made highly accurate.

A random distribution on $\boldsymbol{S} \in \mathbb{R}^{c\times n}$ has independent rows, each row is $\frac{1}{\sqrt{c}}$ times a sample of $\boldsymbol{s} \in \mathbb{R}^n$, then

$$\mathbb{E}[\boldsymbol{S}^\top \boldsymbol{S}] = \mathbb{E}[\sum_{i\in[c]} \boldsymbol{S}_{i*}^\top \boldsymbol{S}_{i*}] = \sum_{i\in[c]} \mathbb{E}[\boldsymbol{S}_{i*}^\top \boldsymbol{S}_{i*}] = \sum_{i\in[c]} \frac{1}{c}\boldsymbol{I} = \boldsymbol{I},$$

so for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we have $\mathbb{E}[\langle \boldsymbol{S}\boldsymbol{x}, \boldsymbol{S}\boldsymbol{y}\rangle] = \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{y}] = \boldsymbol{x}^\top \mathbb{E}[\boldsymbol{S}^\top \boldsymbol{S}]\boldsymbol{y} = \boldsymbol{x}^\top \boldsymbol{y}$.
In particular, $\mathbb{E}[\|\boldsymbol{S}\boldsymbol{y}\|^2] = \|\boldsymbol{y}\|^2$

Applications:

- Approximating matrix multiplication
- Least squares regression
- Low rank approximation

Approximating Matrix Multiplication (AMM)

Problem Statement:

Given an $m \times n$ matrix $\boldsymbol{A}$ and an $n \times p$ matrix $\boldsymbol{B}$, approximate the product $\boldsymbol{A} \cdot \boldsymbol{B}$,

OR, equivalently,

Approximate the sum of n rank-one matrices.

$$\boldsymbol{A} \cdot \boldsymbol{B} = \sum_{i=1}^{n} \underbrace{\begin{bmatrix} \\ \boldsymbol{A}_{*i} \\ \\ \end{bmatrix} \cdot \begin{bmatrix} \quad \boldsymbol{B}_{i*} \quad \end{bmatrix}}_{m \times p}$$

where $\boldsymbol{A}_{*i}$ is the ith column of $\boldsymbol{A}$ and $\boldsymbol{B}_{i*}$ is the ith row of $\boldsymbol{B}$.

Sampling rows of a matrix

- If $\boldsymbol{S} \in \mathbb{R}^{c\times n}$ is a random row sampling matrix, then $\boldsymbol{SA}$:

$$\begin{bmatrix} 0 & \mathrm{s}_{12} & 0 & 0 & \cdots & 0 \\ \mathrm{s}_{21} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \mathrm{s}_{33} & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & \mathrm{s}_{cn} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{1*} \\ \boldsymbol{A}_{2*} \\ \vdots \\ \boldsymbol{A}_{n*} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_{12}\boldsymbol{A}_{2*} \\ \mathrm{s}_{21}\boldsymbol{A}_{1*} \\ \mathrm{s}_{33}\boldsymbol{A}_{3*} \\ \vdots \\ \mathrm{s}_{cn}\boldsymbol{A}_{n*} \end{bmatrix}$$

- As above, for a single sampling vector $\boldsymbol{s}$, *uniform sampling* would pick $i \in [n]$ uniformly with probability $\frac{1}{n}$ and set $\mathrm{s}_i \leftarrow \sqrt{n}$.
- Generally, given $\boldsymbol{p} \in [0,1]^n, \sum_i p_i = 1$. Pick $i \in [n]$ with probability p_i, $\mathrm{s}_i \leftarrow \sqrt{1/p_i}$. We have $\mathbb{E}[\mathrm{s}_i^2] = p_i\sqrt{1/p_i}^2 + (1+p_i)0 = 1$.
- In some instances, by choosing appropriate p_i's, we can get improved results.

AMM - Sampling

$$\boldsymbol{A} \cdot \boldsymbol{B} = \sum_{i=1}^{n} \underbrace{\begin{bmatrix} \boldsymbol{A}_{*i} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{B}_{i*} \end{bmatrix}}_{m \times p}$$

$$\approx \frac{1}{c} \sum_{t=1}^{c} \frac{1}{p_{j_t}} \underbrace{\begin{bmatrix} \boldsymbol{A}_{*j_t} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{B}_{j_t*} \end{bmatrix}}_{m \times p}$$

Pick c terms of the sum, with replacement, with respect to the p_i's. I.e. set $j_t = i$, where $\Pr(j_t = i) = p_i$.

$$\underbrace{\begin{bmatrix} \\ \quad \boldsymbol{A} \quad \\ \\ \end{bmatrix}}_{m\times n} \cdot \underbrace{\begin{bmatrix} \\ \quad \boldsymbol{B} \quad \\ \\ \end{bmatrix}}_{n\times p} \approx \underbrace{\begin{bmatrix} \\ \quad \boldsymbol{C} \quad \\ \\ \end{bmatrix}}_{m\times c} \cdot \underbrace{\begin{bmatrix} \quad \boldsymbol{R} \quad \end{bmatrix}}_{c\times p}$$

- We would like to estimate $\boldsymbol{A}\boldsymbol{B} \approx \boldsymbol{A}\boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{B}$.
- Suppose $\boldsymbol{S}$ has just one row $\boldsymbol{s}_i$. Then, we just get $\boldsymbol{A}_{i*}\mathrm{s}_i^2\boldsymbol{B}_{*i} = \boldsymbol{A}_{*i}\boldsymbol{B}_{i*}/p_i$ with probability p_i.
- If we pick uniformly with $p_i = 1/n$, and suppose one of the row norms $\|\boldsymbol{B}_{1*}\|^2$ is much $\gg$ norms of other rows, then the estimate will be poor, if we miss the row $i = 1$.
- One idea : catch the rows with large norms by setting $p_i \propto \|\boldsymbol{B}_{1*}\|^2$. This is called **Length-squared sampling.**

$$\underbrace{\begin{bmatrix} \\ \quad \boldsymbol{A} \quad \\ \\ \end{bmatrix}}_{m\times n} \cdot \underbrace{\begin{bmatrix} \\ \quad \boldsymbol{B} \quad \\ \\ \end{bmatrix}}_{n\times p} \approx \underbrace{\begin{bmatrix} \\ \quad \boldsymbol{C} \quad \\ \\ \end{bmatrix}}_{m\times c} \cdot \underbrace{\begin{bmatrix} \quad \boldsymbol{R} \quad \end{bmatrix}}_{c\times p}$$

- Create $\boldsymbol{C}$ and $\boldsymbol{R}$ by picking columns $\boldsymbol{A}_{*j_t}$ and rows $\boldsymbol{B}_{j_t*}$ with probability

$$\Pr(j_t = i) = \frac{\|\boldsymbol{A}_{*i}\|_2\|\boldsymbol{B}_{i*}\|_2}{\sum_{j=1}^{n}\|\boldsymbol{A}_{*j}\|_2\|\boldsymbol{B}_{i*}\|_2}$$

- Include $\boldsymbol{A}_{*j_t}/\sqrt{cp_{j_t}}$ as a column of $\boldsymbol{C}$, and $\boldsymbol{B}_{j_t*}/\sqrt{cp_{j_t}}$ as a row of $\boldsymbol{R}$.

Length-squared sampling

Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times p}$. Let $\boldsymbol{S} \in \mathbb{R}^{c\times n}$ be the length squared sampling matrix. Then, $\mathbb{E}[\boldsymbol{C}\boldsymbol{R}] = \boldsymbol{A}\boldsymbol{B}$ (unbiased estimator), where $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{S}^\top$, $\boldsymbol{R} = \boldsymbol{S}\boldsymbol{B}$, and

$$\mathbb{E}[\|\boldsymbol{C}\boldsymbol{R} - \boldsymbol{A}\boldsymbol{B}\|_F^2] \le \frac{1}{c}\|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2$$

Length-squared sampling

Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times p}$. Let $\boldsymbol{S} \in \mathbb{R}^{c\times n}$ be the length squared sampling matrix. Then, $\mathbb{E}[\boldsymbol{C}\boldsymbol{R}] = \boldsymbol{A}\boldsymbol{B}$ (unbiased estimator), where $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{S}^\top, \boldsymbol{R} = \boldsymbol{S}\boldsymbol{B}$, and

$$\mathbb{E}[\|\boldsymbol{C}\boldsymbol{R} - \boldsymbol{A}\boldsymbol{B}\|_F^2] \leq \frac{1}{c}\|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2$$

Proof: First, for any probability p_i, we know that $\mathbb{E}[\boldsymbol{C}\boldsymbol{R}_{ij}] = \boldsymbol{A}\boldsymbol{B}_{ij}$. Elementwise is an unbiased estimator.

Next, note that for a single vector $\boldsymbol{s}, \mathbb{E}[\|\boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{B} - \boldsymbol{A}\boldsymbol{B}\|_F^2]$ is the sum of entry-wise variances.

Since $\mathrm{Var}[\mathrm{x}] = \mathbb{E}[\mathrm{x}^2] - \mathbb{E}[\mathrm{x}]^2$, we have $\mathbb{E}[\|\boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{B} - \boldsymbol{A}\boldsymbol{B}\|_F^2] \leq \mathbb{E}[\|\boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{B}\|_F^2]$

$$
\begin{aligned}
\mathbb{E}[\|\boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{B}\|_F^2] &= \sum_{j,k}\mathbb{E}[(\boldsymbol{A}_{j*}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{B}_{*k})^2] = \sum_{j,k}\mathbb{E}[(\sum_i a_{ji}\mathrm{s}_i^2 b_{ik})^2] \\
&= \sum_{j,k}\sum_i a_{ji}^2 p_i \frac{1}{p_i^2} b_{ik}^2 = \sum_i\sum_j a_{ji}^2\frac{1}{p_i}\sum_k b_{ik}^2 = \sum_i \|\boldsymbol{A}_{*i}\|^2\frac{1}{p_i}\|\boldsymbol{B}_{i*}\|^2 \\
&= \|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2.
\end{aligned}
$$

Next, for the case of c rows, the expected Frobenius norm error is sum of variance of the form

$$
\mathrm{Var}[\sum_{i\in[c]}\mathrm{x}^{(i)}/c] = \sum_{i\in[c]}\mathrm{Var}[\mathrm{x}^{(i)}/c] = \mathrm{Var}[\mathrm{x}^{(1)}]/c.
$$

Thus, we get the result

$$
\mathbb{E}[\|\boldsymbol{C}\boldsymbol{R}-\boldsymbol{A}\boldsymbol{B}\|_F^2] \le \frac{1}{c}\|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2.
$$

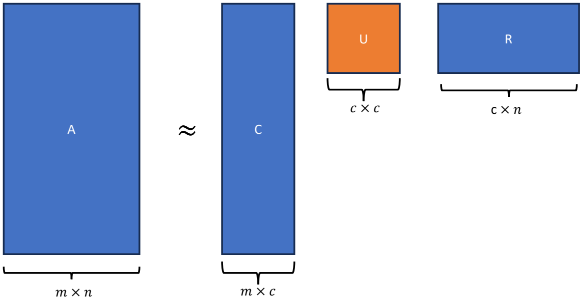
Using Markov's inequality, we can show that for $c \ge 1/\epsilon^2\delta$,

$$
\Pr(\|\boldsymbol{C}\boldsymbol{R}-\boldsymbol{A}\boldsymbol{B}\|_F \ge \epsilon\|\boldsymbol{A}\|_F\|\boldsymbol{B}\|_F) \le \delta.
$$

CUR decomposition

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, a particular type of low rank approximation:

- A row sampling matrix $\boldsymbol{S}_1 \in \mathbb{R}^{c \times m}$, and $\boldsymbol{R} = \boldsymbol{S}_1 \boldsymbol{A} \in \mathbb{R}^{c \times n}$
- A column sampling matrix $\boldsymbol{S}_2 \in \mathbb{R}^{n \times c}$, and $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{S}_2 \in \mathbb{R}^{m \times c}$
- A matrix $\boldsymbol{U} \in \mathbb{R}^{c \times c}$, such that $\boldsymbol{A} \approx \boldsymbol{CUR}$ and $c \ll \{m, n\}$.



CUR decomposition

- We can compute $\boldsymbol{U} = (\boldsymbol{A}\boldsymbol{S}_2)^\dagger \boldsymbol{S}_1^\top = (\boldsymbol{C}^\top \boldsymbol{C})^{-1} (\boldsymbol{S}_1 \boldsymbol{A} \boldsymbol{S}_2)^\top$.
- $\boldsymbol{U}$ can be ill-conditioned.
- Typically, in applications, we are interested in random columns $\boldsymbol{C}$ and rows $\boldsymbol{R}$ of $\boldsymbol{A}$.
- We can also consider, $\boldsymbol{S}_1 \in \mathbb{R}^{r \times m}$ and $\boldsymbol{S}_2 \in \mathbb{R}^{n \times c}$, for different c, r.

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, row sampler $\boldsymbol{S}_1 \in \mathbb{R}^{r \times m}$, column $\boldsymbol{S}_2 \in \mathbb{R}^{n \times c}$, and with $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{S}_2, \boldsymbol{R} = \boldsymbol{S}_1 \boldsymbol{A}, \boldsymbol{U} = (\boldsymbol{A}\boldsymbol{S}_2)^\dagger \boldsymbol{S}_1^\top$, then

$$\mathbb{E}[\|\boldsymbol{C}\boldsymbol{U}\boldsymbol{R} - \boldsymbol{A}\|_2^2] \leq 2\|\boldsymbol{A}\|_F^2 \left(\frac{1}{\sqrt{c}} + \frac{c}{r} \right) \leq \epsilon \|\boldsymbol{A}\|_F^2,$$

for $c = 16/\epsilon^2, r = 64/\epsilon^3$.

Matrix (low rank) approximations

- We can also consider sampling only the columns as $\boldsymbol{A} \approx \boldsymbol{C}\boldsymbol{X}$, or
- Sample only the rows $\boldsymbol{A} \approx \boldsymbol{X}\boldsymbol{R}$.
- More flexible structure can give better-conditioned $\boldsymbol{X}$.
- We need fast decaying spectrum.
- For
$$\Pr(\|\boldsymbol{C}\boldsymbol{U}\boldsymbol{R} - \boldsymbol{A}\|_2 \geq \epsilon\|\boldsymbol{A}\|_F) \leq \delta,$$
we need $c = O(\delta^{-2}\epsilon^{-4}), r = O(\delta^{-3}\epsilon^{-6})$.
- Cost $=?$

Better variance reduction

- We want $\boldsymbol{S}$ such that $\|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x}\|$ is a good estimator of $\|\boldsymbol{A}\boldsymbol{x}\|$.
- Length-squared sampling : $p_i \propto \|\boldsymbol{A}_{i*}\|^2$ is good, but for some $\boldsymbol{x}$, we could have $\boldsymbol{A}_{i*}\boldsymbol{x} = 0$ even if $\|\boldsymbol{A}_{i*}\|^2$ is large.
- We want $(\frac{1}{\sqrt{p_i}}\boldsymbol{A}_{i*}\boldsymbol{x})^2$ to be "well-behaved" for all i and $\boldsymbol{x}$.
- "well-behaved" in one sense : bounded relative contribution to $\|\boldsymbol{A}\boldsymbol{x}\|^2 = \sum_i (\boldsymbol{A}_{i*}\boldsymbol{x})^2$.
- sampling using information related to $span(\boldsymbol{A})$.

Leverage scores

- **Leverage scores:** Given a linear subspace $L \subset \mathbb{R}^m$, for $i \in [m]$, the ith *leverage score* $\ell_i(L) = \sup_{\boldsymbol{y} \in L} y_i^2 / \|\boldsymbol{y}\|^2$.
- The leverage scores of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are $\ell_i(\boldsymbol{A}) = \ell_i(span(\boldsymbol{A}))$.

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and an orthonormal basis $\boldsymbol{U}$ for $span(\boldsymbol{A})$, for $i \in [m]$, the ith *leverage score*

$$\ell_i(\boldsymbol{A}) = \sup_{\boldsymbol{x}} \frac{(\boldsymbol{A}_{i*}\boldsymbol{x})^2}{\|\boldsymbol{A}\boldsymbol{x}\|^2} = \|\boldsymbol{U}_{i*}\|^2.$$

Leverage scores

- **Leverage scores:** Given a linear subspace $L \subset \mathbb{R}^m$, for $i \in [m]$, the ith *leverage score* $\ell_i(L) = \sup_{\boldsymbol{y} \in L} y_i^2 / \|\boldsymbol{y}\|^2$.
- The leverage scores of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are $\ell_i(\boldsymbol{A}) = \ell_i(span(\boldsymbol{A}))$.

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and an orthonormal basis $\boldsymbol{U}$ for $span(\boldsymbol{A})$, for $i \in [m]$, the ith *leverage score*

$$\ell_i(\boldsymbol{A}) = \sup_{\boldsymbol{x}} \frac{(\boldsymbol{A}_{i*}\boldsymbol{x})^2}{\|\boldsymbol{A}\boldsymbol{x}\|^2} = \|\boldsymbol{U}_{i*}\|^2.$$

For $L = span(\boldsymbol{A}) = span(\boldsymbol{U})$, and $\boldsymbol{z} \in L$ has $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{U}\boldsymbol{y}$ for some $\boldsymbol{x}, \boldsymbol{y}$. So,

$$\sup_{\boldsymbol{x}} \frac{(\boldsymbol{A}_{i*}\boldsymbol{x})^2}{\|\boldsymbol{A}\boldsymbol{x}\|^2} = \sup_{\boldsymbol{y}} \frac{(\boldsymbol{U}_{i*}\boldsymbol{y})^2}{\|\boldsymbol{U}\boldsymbol{y}\|^2} = \sup_{\boldsymbol{y}} \frac{(\boldsymbol{U}_{i*}\boldsymbol{y})^2}{\|\boldsymbol{y}\|^2} = \|\boldsymbol{U}_{i*}\|^2.$$

We have $\ell_i(\boldsymbol{A}) \in [0, 1]$ and $\sum_i \ell_i(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A})$.

Leverage score sampling

Leverage score sampling: sample rows with probability proportional to the square of the Euclidean norms of the rows of the left singular vectors of $\boldsymbol{A}$.

$$p_i = \frac{\|\boldsymbol{U}_{i*}\|^2}{\|\boldsymbol{U}\|_F^2} = \frac{\|\boldsymbol{U}_{i*}\|^2}{n}$$

Column sampling is equivalent to row sampling by focusing on $\boldsymbol{A}^\top$. So, we consider the right singular vectors $\boldsymbol{V}$.

$$p_j = \frac{\|\boldsymbol{V}_{j*}\|^2}{n}.$$

Leverage scores: general case

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{A}_k$ its best rank-k approximation (as computed by the SVD):

$$\boldsymbol{A} \approx \underbrace{\begin{bmatrix} & & \\ & \boldsymbol{A}_k & \\ & & \end{bmatrix}}_{m\times n} \approx \underbrace{\begin{bmatrix} & & \\ & \boldsymbol{U}_k & \\ & & \end{bmatrix}}_{m\times k} \cdot \underbrace{\begin{bmatrix} & \Sigma_k & \end{bmatrix}}_{k\times k} \cdot \underbrace{\begin{bmatrix} & \boldsymbol{V}_k^\top & \end{bmatrix}}_{k\times n}$$

Row Leverage scores and *Column Leverage scores*

$$p_i = \frac{\|(\boldsymbol{U}_k)_{i*}\|^2}{k} \qquad\qquad p_j = \frac{\|(\boldsymbol{V}_k)_{j*}\|^2}{k}$$

Leverage score sampling

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, if we randomly sample the columns $\boldsymbol{C} \in \mathbb{R}^{m \times c}$ using leverage scores, then, with probability at least 0.9,

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{X}\|_F = \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^\dagger \boldsymbol{A}\|_F \leq (1+\epsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F,$$

for sampling complexity

$$c = O\left(\frac{k}{\epsilon^2} \log\left(\frac{k}{\epsilon}\right)\right)$$

Leverage score sampling

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, if we randomly sample the columns $\boldsymbol{C} \in \mathbb{R}^{m \times c}$ using leverage scores, then, with probability at least 0.9,

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{X}\|_F = \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^\dagger \boldsymbol{A}\|_F \leq (1+\epsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F,$$

for sampling complexity

$$c = O\left(\frac{k}{\epsilon^2} \log\left(\frac{k}{\epsilon}\right)\right)$$

Proof uses Matrix Chernoff inequality.
Let $\boldsymbol{X}_i$ for $i \in [c]$ be i.i.d copies of symmetric random $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ with $\gamma, \sigma^2 > 0$, $\mathbb{E}[\boldsymbol{X}] = 0$, $\|\boldsymbol{X}\|_2 \leq \gamma$, and $\|\mathbb{E}[\boldsymbol{X}^2]\|_2 \leq \sigma^2$. Then for $\epsilon > 0$,

$$\Pr(\|\frac{1}{c}\sum_i \boldsymbol{X}_i\|_2 \geq \epsilon) \leq 2n\exp(-c\epsilon^2/(\sigma^2 + \gamma\epsilon/3)).$$

Further Reading

- Drineas, Petros, Ravi Kannan, and Michael W. Mahoney. “Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication.” SIAM Journal on Computing 36.1 (2006): 132-157.
- Drineas, Petros, Ravi Kannan, and Michael W. Mahoney. “Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix.” SIAM Journal on computing 36.1 (2006): 158-183.
- Kannan, Ravindran, and Santosh Vempala. “Randomized algorithms in numerical linear algebra.” Acta Numerica 26 (2017): 95-135.
- Boutsidis, Christos, and David P. Woodruff. “Optimal CUR matrix decompositions.” Proceedings of the forty-sixth annual ACM symposium on Theory of computing. 2014.

Questions?