# Computations with Low Rank Matrices in Logarithmic Space

Attila Pereszlényi[1] peresz@sztaki.hu
Joint work with Katalin Friedl[2], László Kabódi[2], Dániel Szabó[2]

[1]Institute for Computer Science and Control, Budapest, Hungary (SZTAKI)

[2]Department of Computer Science and Information Theory (BME SZIT)

Mathematical Modeling Seminar 3rd November, 2020

# Outline



SZTAKI

# Outline



SZTAKI

# Outline

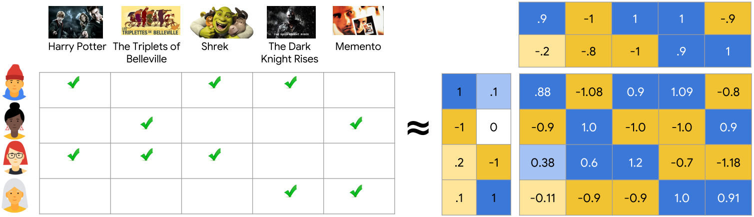
# Why low rank matrices?

Matrices that are low rank or can be approximated well by low rank matrices are common in many areas.

# Why low rank matrices?

Matrices that are low rank or can be approximated well by low rank matrices are common in many areas.

- Example: recommendation systems




Source: Google `developers.google.com/machine-learning/recommendation/collaborative/matrix`

# Singular value decomposition

The singular value decomposition (SVD) of $M \in \mathbb{R}^{m \times n}$ is

$$M = U \Sigma V^{\mathsf{T}}$$

# Singular value decomposition

The singular value decomposition (SVD) of $\mathbf{M} \in \mathbb{R}^{m \times n}$ is

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$$

- $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$ where $r = \text{rank}(\mathbf{M})$

# Singular value decomposition

The singular value decomposition (SVD) of $\mathbf{M} \in \mathbb{R}^{m \times n}$ is

$$\mathbf{M} = \mathbf{U} \Sigma \mathbf{V}^\mathsf{T}$$

- $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$ where $r = \mathrm{rank}(\mathbf{M})$
- $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbb{1}$

SZTAKI

# Singular value decomposition

The singular value decomposition (SVD) of $\mathbf{M} \in \mathbb{R}^{m \times n}$ is

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$$

- $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$ where $r = \operatorname{rank}(\mathbf{M})$
- $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbb{1}$
- $\Sigma = \operatorname{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$
  where $\sigma_1(\mathbf{M}) \geqslant \sigma_2(\mathbf{M}) \geqslant \ldots \geqslant \sigma_r(\mathbf{M}) > 0$

# Singular value decomposition

The singular value decomposition (SVD) of $M \in \mathbb{R}^{m \times n}$ is

$$M = U \Sigma V^T$$

- $U \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $V \in \mathbb{R}^{n \times r}$ where $r = \text{rank}(M)$
- $U^T U = V^T V = \mathbb{1}$
- $\Sigma = \text{diag}(\sigma_1(M), \sigma_2(M), \ldots, \sigma_r(M))$ where $\sigma_1(M) \geqslant \sigma_2(M) \geqslant \ldots \geqslant \sigma_r(M) > 0$
- The $\sigma_i(M)$'s are called the singular values of $M$ and the columns of $U$ and $V$ are called the left and right singular vectors of $M$.

# Some notations

- We denote the
  - $(i, j)$-th element of $\mathbf{M}$ by $\mathbf{M}(i, j)$

# Some notations

- We denote the
  - $(i, j)$-th element of $\mathbf{M}$ by $\mathbf{M}(i, j)$
  - $i$-th row of $\mathbf{M}$ by $\mathbf{M}(i, :)$

# Some notations

- We denote the
  - $(i, j)$-th element of $M$ by $M(i, j)$
  - $i$-th row of $M$ by $M(i, :)$
  - $j$-th column of $M$ by $M(:, j)$

# Some notations

- We denote the
  - $(i, j)$-th element of $\mathbf{M}$ by $\mathbf{M}(i, j)$
  - $i$-th row of $\mathbf{M}$ by $\mathbf{M}(i, :)$
  - $j$-th column of $\mathbf{M}$ by $\mathbf{M}(:, j)$
- The Frobenius norm of $\mathbf{M}$ is

$$\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{M}(i, j)^2}$$

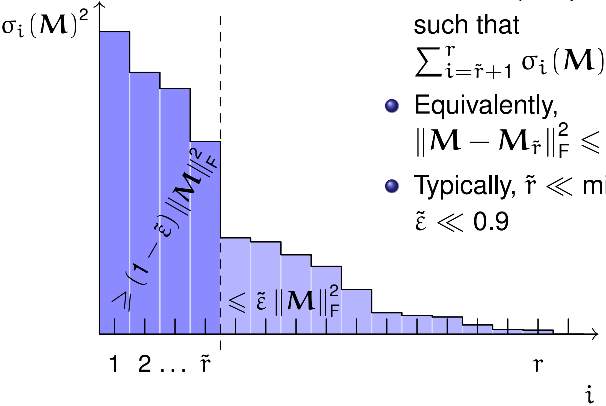
# Some notations

- We denote the
  - $(i, j)$-th element of $M$ by $M(i, j)$
  - $i$-th row of $M$ by $M(i, :)$
  - $j$-th column of $M$ by $M(:, j)$
- The Frobenius norm of $M$ is

$$\|M\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} M(i, j)^2} = \sqrt{\sum_{i=1}^{r} \sigma_i(M)^2}$$

# Approximately low rank matrices

If $\mathbf{M}$ can be approximated well by a low rank matrix then its singular values look like this.




- $\exists \tilde{r} \in \mathbb{Z}^+$, $\tilde{r} \leqslant r$ and $0 \leqslant \tilde{\varepsilon} \leqslant 0.9$ such that $\sum_{i=\tilde{r}+1}^{r} \sigma_i(\mathbf{M})^2 \leqslant \tilde{\varepsilon}\|\mathbf{M}\|_F^2$
- Equivalently, $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F^2 \leqslant \tilde{\varepsilon}\|\mathbf{M}\|_F^2$
- Typically, $\tilde{r} \ll \min\{m, n\}$ and $\tilde{\varepsilon} \ll 0.9$

# Outline

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column.

# Sampling a column

Given $M \in \mathbb{R}^{m \times n}$ let's estimate $MM^T$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let

$$\Pr[J = j] = \frac{\|M(:, j)\|^2}{\|M\|_F^2}$$

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m\times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:,j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:,j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$
- Let $\mathbf{c} = \frac{\mathbf{M}(:,J)}{\sqrt{p_J}}$.

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$
- Let $\mathbf{c} = \frac{\mathbf{M}(:, J)}{\sqrt{p_J}}$. We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{c}\mathbf{c}^\mathsf{T}$.

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$
- Let $\mathbf{c} = \frac{\mathbf{M}(:, J)}{\sqrt{p_J}}$. We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{c}\mathbf{c}^\mathsf{T}$.

$$\mathbb{E}\left[\mathbf{c}\mathbf{c}^\mathsf{T}\right] = \sum_{k=1}^{n} \Pr[J = k] \frac{\mathbf{M}(:, k)\,\mathbf{M}^\mathsf{T}(k, :)}{p_k}$$

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$
- Let $\mathbf{c} = \frac{\mathbf{M}(:, J)}{\sqrt{p_J}}$. We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{c}\mathbf{c}^\mathsf{T}$.

$$\begin{aligned}\mathbb{E}\left[\mathbf{c}\mathbf{c}^\mathsf{T}\right] &= \sum_{k=1}^{n} \Pr[J = k] \frac{\mathbf{M}(:, k)\,\mathbf{M}^\mathsf{T}(k, :)}{p_k} \\ &= \sum_{k=1}^{n} \mathbf{M}(:, k)\,\mathbf{M}^\mathsf{T}(k, :)\end{aligned}$$

# Sampling a column

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$ let's estimate $\mathbf{M}\mathbf{M}^\mathsf{T}$.

- Pick a column randomly. Let $J \in [n]$ be the index of the picked column. Let
$$\Pr[J = j] = \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$
- Let $\mathbf{c} = \frac{\mathbf{M}(:, J)}{\sqrt{p_J}}$. We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{c}\mathbf{c}^\mathsf{T}$.

$$\begin{aligned}
\mathbb{E}\left[\mathbf{c}\mathbf{c}^\mathsf{T}\right] &= \sum_{k=1}^{n} \Pr[J = k] \frac{\mathbf{M}(:, k)\,\mathbf{M}^\mathsf{T}(k, :)}{p_k} \\
&= \sum_{k=1}^{n} \mathbf{M}(:, k)\,\mathbf{M}^\mathsf{T}(k, :) \\
&= \mathbf{M}\mathbf{M}^\mathsf{T}
\end{aligned}$$

# Variance

How good is the approximation?

# Variance

How good is the approximation?

$$\mathrm{Var}\left(\mathbf{c}\mathbf{c}^{\mathsf{T}}\right) \stackrel{\text{def}}{=} \mathbb{E}\left[\left\|\mathbf{c}\mathbf{c}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right]$$

# Variance

How good is the approximation?

$$\mathrm{Var}\left(cc^{\mathsf{T}}\right) \stackrel{\text{def}}{=} \mathbb{E}\left[\left\|cc^{\mathsf{T}} - MM^{\mathsf{T}}\right\|_F^2\right] = \sum_{i,j\in[m]} \mathrm{Var}\left(\left(cc^{\mathsf{T}}\right)(i,j)\right)$$

# Variance

How good is the approximation?

$$\begin{aligned}
\operatorname{Var}\left(\mathbf{c}\mathbf{c}^{\mathsf{T}}\right) &\stackrel{\text{def}}{=} \mathbb{E}\left[\left\|\mathbf{c}\mathbf{c}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \\
&= \sum_{i,j \in [m]} \operatorname{Var}\left(\left(\mathbf{c}\mathbf{c}^{\mathsf{T}}\right)(i,j)\right) \\
&= \sum_{i,j \in [m]} \operatorname{Var}\left(\frac{\mathbf{M}(i,J)\,\mathbf{M}(j,J)}{p_J}\right)
\end{aligned}$$

# Variance

How good is the approximation?

$$\begin{aligned}
\mathrm{Var}\left(cc^{\mathsf{T}}\right) &\stackrel{\text{def}}{=} \mathbb{E}\left[\left\|cc^{\mathsf{T}} - MM^{\mathsf{T}}\right\|_F^2\right] \\
&= \sum_{i,j\in[m]} \mathrm{Var}\left(\left(cc^{\mathsf{T}}\right)(i,j)\right) \\
&= \sum_{i,j\in[m]} \mathrm{Var}\left(\frac{M(i,J)\,M(j,J)}{p_J}\right) \\
&\leqslant \sum_{i,j\in[m]} \mathbb{E}\left[\frac{M(i,J)^2\,M(j,J)^2}{p_J^2}\right]
\end{aligned}$$

# Variance cont.

$$\mathbb{E}\left[\left\|\mathbf{c}\mathbf{c}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \mathbb{E}\left[\sum_{i,j\in[m]} \frac{\mathbf{M}(i,J)^2\,\mathbf{M}(j,J)^2\,\|\mathbf{M}\|_{\mathsf{F}}^4}{\|\mathbf{M}(:,J)\|^4}\right]$$

# Variance cont.

$$\mathbb{E}\left[\left\|\mathbf{c}\mathbf{c}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \mathbb{E}\left[\sum_{i,j\in[m]} \frac{\mathbf{M}(i,J)^{2}\,\mathbf{M}(j,J)^{2}\,\|\mathbf{M}\|_{\mathsf{F}}^{4}}{\|\mathbf{M}(:,J)\|^{4}}\right]$$

$$= \|\mathbf{M}\|_{\mathsf{F}}^{4} \cdot \mathbb{E}\left[\left(\sum_{i=1}^{m} \frac{\mathbf{M}(i,J)^{2}}{\|\mathbf{M}(:,J)\|^{2}}\right)\left(\sum_{j=1}^{m} \frac{\mathbf{M}(j,J)^{2}}{\|\mathbf{M}(:,J)\|^{2}}\right)\right]$$

# Variance cont.

$$\mathbb{E}\left[\left\|\mathbf{c}\mathbf{c}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \mathbb{E}\left[\sum_{i,j \in [m]} \frac{\mathbf{M}(i,J)^2\, \mathbf{M}(j,J)^2\, \|\mathbf{M}\|_{\mathsf{F}}^4}{\|\mathbf{M}(:,J)\|^4}\right]$$

$$= \|\mathbf{M}\|_{\mathsf{F}}^4 \cdot \mathbb{E}\left[\left(\sum_{i=1}^{m} \frac{\mathbf{M}(i,J)^2}{\|\mathbf{M}(:,J)\|^2}\right)\left(\sum_{j=1}^{m} \frac{\mathbf{M}(j,J)^2}{\|\mathbf{M}(:,J)\|^2}\right)\right]$$

$$= \|\mathbf{M}\|_{\mathsf{F}}^4$$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k^\mathsf{T}}{s}\right]$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 \mathbf{c}_1^\mathsf{T}\right]$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 \mathbf{c}_1^\mathsf{T}\right] = \mathbf{M}\mathbf{M}^\mathsf{T}$

SZTAKI

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k {\mathbf{c}_k}^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 {\mathbf{c}_1}^\mathsf{T}\right] = \mathbf{M}\mathbf{M}^\mathsf{T}$
- $\mathrm{Var}\left(\mathbf{C}\mathbf{C}^\mathsf{T}\right) = \mathrm{Var}\left(\sum_{k=1}^{s} \frac{\mathbf{c}_k {\mathbf{c}_k}^\mathsf{T}}{s}\right)$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right] = \mathbf{M}\mathbf{M}^\mathsf{T}$
- $\mathrm{Var}\left(\mathbf{C}\mathbf{C}^\mathsf{T}\right) = \mathrm{Var}\left(\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right) = \frac{\mathrm{Var}\left(\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right)}{s}$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right] = \mathbf{M}\mathbf{M}^\mathsf{T}$
- $\mathrm{Var}\left(\mathbf{C}\mathbf{C}^\mathsf{T}\right) = \mathrm{Var}\left(\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right) = \frac{\mathrm{Var}\left(\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right)}{s} \leqslant \frac{\|\mathbf{M}\|_F^4}{s}$

# Sampling multiple columns

Let's sample $s$ columns independently: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s$.

- Form matrix $\mathbf{C} \in \mathbb{R}^{m \times s}$ by $\mathbf{C}(:, j) \stackrel{\text{def}}{=} \frac{\mathbf{c}_j}{\sqrt{s}}$.
- We approximate $\mathbf{M}\mathbf{M}^\mathsf{T}$ with $\mathbf{C}\mathbf{C}^\mathsf{T}$.
- $\mathbb{E}\left[\mathbf{C}\mathbf{C}^\mathsf{T}\right] = \mathbb{E}\left[\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right] = \mathbb{E}\left[\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right] = \mathbf{M}\mathbf{M}^\mathsf{T}$
- $\mathrm{Var}\left(\mathbf{C}\mathbf{C}^\mathsf{T}\right) = \mathrm{Var}\left(\sum_{k=1}^{s} \frac{\mathbf{c}_k \mathbf{c}_k{}^\mathsf{T}}{s}\right) = \frac{\mathrm{Var}\left(\mathbf{c}_1 \mathbf{c}_1{}^\mathsf{T}\right)}{s} \leqslant \frac{\|\mathbf{M}\|_F^4}{s}$

$$\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^\mathsf{T} - \mathbf{M}\mathbf{M}^\mathsf{T}\right\|_F^2\right] \leqslant \frac{\|\mathbf{M}\|_F^4}{s}$$

SZTAKI

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}$

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}$

- Let $r = \text{rank}(\mathbf{M})$.

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^2\right] \leqslant \frac{\|\mathbf{M}\|_{\mathsf{F}}^4}{s}$

- Let $\mathrm{r} = \mathrm{rank}(\mathbf{M})$.
- Suppose $\mathbf{C} = 0$ (trivial approximation).

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] \leqslant \frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}$

- Let $r = \operatorname{rank}(\mathbf{M})$.
- Suppose $\mathbf{C} = 0$ (trivial approximation).
- Left hand side: $\left\|\mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2} = \sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^{4}$

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^\mathsf{T} - \mathbf{M}\mathbf{M}^\mathsf{T}\right\|_\mathsf{F}^2\right] \leqslant \frac{\|\mathbf{M}\|_\mathsf{F}^4}{s}$

- Let $r = \operatorname{rank}(\mathbf{M})$.
- Suppose $\mathbf{C} = 0$ (trivial approximation).
- Left hand side: $\left\|\mathbf{M}\mathbf{M}^\mathsf{T}\right\|_\mathsf{F}^2 = \sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^4$
- Right hand side: $\frac{1}{s}\left(\sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^2\right)^2$

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_F^2\right] \leqslant \frac{\|\mathbf{M}\|_F^4}{s}$

- Let $r = \operatorname{rank}(\mathbf{M})$.
- Suppose $\mathbf{C} = 0$ (trivial approximation).
- Left hand side: $\left\|\mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_F^2 = \sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^4$
- Right hand side: $\frac{1}{s}\left(\sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^2\right)^2$
- For the bound to be non-trivial:

$$\frac{1}{s}\left(\sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^2\right)^2 < \sum_{k=1}^{r} \sigma_{\mathbf{M}}(k)^4$$

# Why does $\mathbf{M}$ have to be low rank?

We have the following bound: $\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^\mathsf{T} - \mathbf{M}\mathbf{M}^\mathsf{T}\right\|_\mathsf{F}^2\right] \leqslant \frac{\|\mathbf{M}\|_\mathsf{F}^4}{s}$

- Let $r = \mathrm{rank}(\mathbf{M})$.
- Suppose $\mathbf{C} = 0$ (trivial approximation).
- Left hand side: $\left\|\mathbf{M}\mathbf{M}^\mathsf{T}\right\|_\mathsf{F}^2 = \sum_{k=1}^{r} \sigma_\mathbf{M}(k)^4$
- Right hand side: $\frac{1}{s}\left(\sum_{k=1}^{r} \sigma_\mathbf{M}(k)^2\right)^2$
- For the bound to be non-trivial:

$$\frac{1}{s}\left(\sum_{k=1}^{r} \sigma_\mathbf{M}(k)^2\right)^2 < \sum_{k=1}^{r} \sigma_\mathbf{M}(k)^4$$

which holds if $s > r$.

SZTAKI

# Outline

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.
  - A random walk is a random sequence of vertices. We start from a uniform random vertex. In each step we choose a random neighbor of the current vertex.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.
  - A random walk is a random sequence of vertices. We start from a uniform random vertex. In each step we choose a random neighbor of the current vertex.
  - Randomness required for a walk of length $s$: $\log(N) + s\log(d)$. ($N$ is the number of vertices.)

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.
  - A random walk is a random sequence of vertices. We start from a uniform random vertex. In each step we choose a random neighbor of the current vertex.
  - Randomness required for a walk of length $s$: $\log(N) + s\log(d)$. ($N$ is the number of vertices.)
  - A graph is a good expander if, after $O(\log(N))$ steps, the probability to arrive at a given vertex is close to uniform on all vertices.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.
  - A random walk is a random sequence of vertices. We start from a uniform random vertex. In each step we choose a random neighbor of the current vertex.
  - Randomness required for a walk of length $s$: $\log(N) + s \log(d)$. ($N$ is the number of vertices.)
  - A graph is a good expander if, after $O(\log(N))$ steps, the probability to arrive at a given vertex is close to uniform on all vertices. Also, $d = O(1)$.

# Random walks on expander graphs

- Sampling $s$ columns independently requires $s \cdot \log(n)$ random bits.
- Perform random walk on an expander graph to reduce randomness.
  - We only consider graphs that are undirected, connected, $d$-regular, non-bipartite.
  - A random walk is a random sequence of vertices. We start from a uniform random vertex. In each step we choose a random neighbor of the current vertex.
  - Randomness required for a walk of length $s$: $\log(N) + s \log(d)$. ($N$ is the number of vertices.)
  - A graph is a good expander if, after $O(\log(N))$ steps, the probability to arrive at a given vertex is close to uniform on all vertices. Also, $d = O(1)$.
  - We need: good expanders exist for all $N = n^4$.

SZTAKI

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.
- Label the vertices with columns of $\mathbf{M}$. Formally, take $f : [N] \to [n]$

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.
- Label the vertices with columns of $\mathbf{M}$. Formally, take $f : [N] \to [n]$ such that for a uniform $v \in [N]$

$$\Pr[f(v) = j] \approx \frac{\|\mathbf{M}(:,j)\|^2}{\|\mathbf{M}\|_F^2}$$

SZTAKI

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.
- Label the vertices with columns of $\mathbf{M}$. Formally, take $f : [N] \to [n]$ such that for a uniform $v \in [N]$

$$\Pr[f(v) = j] \approx \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.
- Label the vertices with columns of $\mathbf{M}$. Formally, take $f : [N] \to [n]$ such that for a uniform $v \in [N]$

$$\Pr[f(v) = j] \approx \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$

- Let $v_1, v_2, \ldots, v_s$ be the vertices visited by the walk. Then

$$\mathbf{C} = \frac{1}{\sqrt{s}} \left[ \frac{\mathbf{M}(:, f(v_1))}{\sqrt{p_{f(v_1)}}} \quad \frac{\mathbf{M}(:, f(v_2))}{\sqrt{p_{f(v_2)}}} \quad \ldots \quad \frac{\mathbf{M}(:, f(v_s))}{\sqrt{p_{f(v_s)}}} \right].$$

# Labeling vertices with columns

- Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, take an expander graph with $N = n^4$ vertices.
- Label the vertices with columns of $\mathbf{M}$. Formally, take $f : [N] \to [n]$ such that for a uniform $v \in [N]$

$$\Pr[f(v) = j] \approx \frac{\|\mathbf{M}(:, j)\|^2}{\|\mathbf{M}\|_F^2} = p_j.$$

- Let $v_1, v_2, \ldots, v_s$ be the vertices visited by the walk. Then

$$\mathbf{C} = \frac{1}{\sqrt{s}} \left[ \frac{\mathbf{M}(:, f(v_1))}{\sqrt{p_{f(v_1)}}} \quad \frac{\mathbf{M}(:, f(v_2))}{\sqrt{p_{f(v_2)}}} \quad \cdots \quad \frac{\mathbf{M}(:, f(v_s))}{\sqrt{p_{f(v_s)}}} \right].$$

- Randomness required: $O(\log(n) + s)$.

SZTAKI

# Approximation with expanders

**Theorem**

*For $\mathbf{M} \in \mathbb{R}^{m \times n}$, let $\mathbf{C} \in \mathbb{R}^{m \times s}$ be the matrix we get by the sampling procedure. It holds that*

$$\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}\right).$$

# Approximation with expanders

## Theorem

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *let* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *be the matrix we get by the sampling procedure. It holds that*

$$\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}\right).$$

*This procedure uses* $O(s + \log(mn))$ *space including the number of random bits.*

# Approximation with expanders

## Theorem

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$*, let* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *be the matrix we get by the sampling procedure. It holds that*

$$\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}\right).$$

*This procedure uses* $O(s + \log(mn))$ *space including the number of random bits.*

- The constant in the big $O$ notation only depends on the expansion parameter of the graph.

# Approximation with expanders

## Theorem

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$*, let* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *be the matrix we get by the sampling procedure. It holds that*

$$\mathbb{E}\left[\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}}^{2}\right] = \mathrm{O}\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{4}}{s}\right).$$

*This procedure uses* $\mathrm{O}(s + \log(mn))$ *space including the number of random bits.*

- The constant in the big $\mathrm{O}$ notation only depends on the expansion parameter of the graph.
- Only the "work space" counts in the space complexity, reading the input and writing the output does not.

# Derandomized algorithm

We can iterate over all the possible random bits.

# Derandomized algorithm

We can iterate over all the possible random bits.

## Theorem

*There exists a deterministic algorithm that, on input* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *outputs* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *for which*

$$\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}} = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{2}}{\sqrt{s}}\right).$$

# Derandomized algorithm

We can iterate over all the possible random bits.

**Theorem**

*There exists a deterministic algorithm that, on input* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *outputs* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *for which*

$$\|\mathbf{C}\mathbf{C}^\mathsf{T} - \mathbf{M}\mathbf{M}^\mathsf{T}\|_\mathsf{F} = O\left(\frac{\|\mathbf{M}\|_\mathsf{F}^2}{\sqrt{s}}\right).$$

*The algorithm uses* $O(s + \log(mn))$ *space.*

# Derandomized algorithm

We can iterate over all the possible random bits.

**Theorem**

*There exists a deterministic algorithm that, on input* $\mathbf{M} \in \mathbb{R}^{m \times n}$*, outputs* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *for which*

$$\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}} = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^2}{\sqrt{s}}\right).$$

*The algorithm uses* $O(s + \log(mn))$ *space.*

- $\left\|\mathbf{A}\mathbf{A}^{\mathsf{T}} - \mathbf{B}\mathbf{B}^{\mathsf{T}}\right\|_{\mathsf{F}}$ is a distance measure between $\mathbf{A}$ and $\mathbf{B}$.

# Derandomized algorithm

We can iterate over all the possible random bits.

**Theorem**

*There exists a deterministic algorithm that, on input* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *outputs* $\mathbf{C} \in \mathbb{R}^{m \times s}$ *for which*

$$\left\|\mathbf{C}\mathbf{C}^{\mathsf{T}} - \mathbf{M}\mathbf{M}^{\mathsf{T}}\right\|_{\mathsf{F}} = O\left(\frac{\|\mathbf{M}\|_{\mathsf{F}}^{2}}{\sqrt{s}}\right).$$

*The algorithm uses* $O(s + \log(mn))$ *space.*

- $\left\|\mathbf{A}\mathbf{A}^{\mathsf{T}} - \mathbf{B}\mathbf{B}^{\mathsf{T}}\right\|_{\mathsf{F}}$ is a distance measure between $\mathbf{A}$ and $\mathbf{B}$.
- For different random bits we get different $\mathbf{C}\mathbf{C}^{\mathsf{T}}$ matrices. We just have to pick one that is close to many others.

# Outline

# Low rank approximation with singular vectors

## Theorem (known from earlier)

*For any* $\mathbf{A} \in \mathbb{R}^{m \times s}$, $\mathbf{M} \in \mathbb{R}^{m \times n}$, $k \geqslant 1$, *if the columns of* $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ *are the left singular vectors corresponding to the top* $k$ *singular values of* $\mathbf{A}$ *and* $\mathbf{M}_k$ *is the best rank-*$k$ *approximation to* $\mathbf{M}$ *then*

$$\left\|\mathbf{M} - \mathbf{U}_k \mathbf{U}_k^T \mathbf{M}\right\|_F^2 \leqslant \|\mathbf{M} - \mathbf{M}_k\|_F^2 + 2\sqrt{k}\left\|\mathbf{A}\mathbf{A}^T - \mathbf{M}\mathbf{M}^T\right\|_F .$$

# Low rank approximation with singular vectors

**Theorem (known from earlier)**

*For any* $\mathbf{A} \in \mathbb{R}^{m \times s}$, $\mathbf{M} \in \mathbb{R}^{m \times n}$, $k \geqslant 1$, *if the columns of* $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ *are the left singular vectors corresponding to the top* $k$ *singular values of* $\mathbf{A}$ *and* $\mathbf{M}_k$ *is the best rank-*$k$ *approximation to* $\mathbf{M}$ *then*

$$\left\|\mathbf{M} - \mathbf{U}_k\mathbf{U}_k^T\mathbf{M}\right\|_F^2 \leqslant \|\mathbf{M} - \mathbf{M}_k\|_F^2 + 2\sqrt{k}\left\|\mathbf{A}\mathbf{A}^T - \mathbf{M}\mathbf{M}^T\right\|_F.$$

- $\mathbf{U}_k^T\mathbf{U}_k = \mathbb{1}$ while $\mathbf{U}_k\mathbf{U}_k^T \leqslant \mathbb{1}$ is the projection to the range of the best rank-$k$ approximation to $\mathbf{A}$.

# Low rank approximation with singular vectors

**Theorem (known from earlier)**

*For any $\mathbf{A} \in \mathbb{R}^{m \times s}$, $\mathbf{M} \in \mathbb{R}^{m \times n}$, $k \geqslant 1$, if the columns of $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ are the left singular vectors corresponding to the top $k$ singular values of $\mathbf{A}$ and $\mathbf{M}_k$ is the best rank-$k$ approximation to $\mathbf{M}$ then*

$$\left\|\mathbf{M} - \mathbf{U}_k \mathbf{U}_k^T \mathbf{M}\right\|_F^2 \leqslant \|\mathbf{M} - \mathbf{M}_k\|_F^2 + 2\sqrt{k}\left\|\mathbf{A}\mathbf{A}^T - \mathbf{M}\mathbf{M}^T\right\|_F .$$

- $\mathbf{U}_k^T \mathbf{U}_k = \mathbb{1}$ while $\mathbf{U}_k \mathbf{U}_k^T \leqslant \mathbb{1}$ is the projection to the range of the best rank-$k$ approximation to $\mathbf{A}$.
- The first term is the best possible error.

# Low rank approximation with singular vectors

## Theorem (known from earlier)

*For any* $A \in \mathbb{R}^{m \times s}$, $M \in \mathbb{R}^{m \times n}$, $k \geqslant 1$, *if the columns of* $U_k \in \mathbb{R}^{m \times k}$ *are the left singular vectors corresponding to the top* $k$ *singular values of* $A$ *and* $M_k$ *is the best rank-*$k$ *approximation to* $M$ *then*

$$\|M - U_k U_k^T M\|_F^2 \leqslant \|M - M_k\|_F^2 + 2\sqrt{k}\,\|AA^T - MM^T\|_F .$$

- $U_k^T U_k = \mathbb{1}$ while $U_k U_k^T \leqslant \mathbb{1}$ is the projection to the range of the best rank-$k$ approximation to $A$.
- The first term is the best possible error.
- The second term is what we pay for calculating with $A$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^{\mathsf{T}}\mathbf{C} \in \mathbb{R}^{s \times s}$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^\mathsf{T}\mathbf{C} \in \mathbb{R}^{s \times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^+$ instead of $\mathbf{U}_k\mathbf{U}_k^\mathsf{T}$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^{\mathsf{T}}\mathbf{C} \in \mathbb{R}^{s \times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^{+}$ instead of $\mathbf{U}_k\mathbf{U}_k^{\mathsf{T}}$.
  - $\mathbf{C}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^{\mathsf{T}}\mathbf{C} \in \mathbb{R}^{s \times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^{+}$ instead of $\mathbf{U}_k\mathbf{U}_k^{\mathsf{T}}$.
  - $\mathbf{C}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.
  - If the SVD of $\mathbf{C}$ is $\mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$ then $\mathbf{C}^{+} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{\mathsf{T}}$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\mathbf{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^\mathsf{T}\mathbf{C} \in \mathbb{R}^{s\times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^+$ instead of $\mathbf{U}_k\mathbf{U}_k^\mathsf{T}$.
    - $\mathbf{C}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.
    - If the SVD of $\mathbf{C}$ is $\mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$ then $\mathbf{C}^+ = \mathbf{V}\Sigma^{-1}\mathbf{U}^\mathsf{T}$.
    - So, $\mathbf{C}\mathbf{C}^+ = \mathbf{U}\mathbf{U}^\mathsf{T}$

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^\mathsf{T}\mathbf{C} \in \mathbb{R}^{s\times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^{+}$ instead of $\mathbf{U}_k\mathbf{U}_k^\mathsf{T}$.
  - $\mathbf{C}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.
  - If the SVD of $\mathbf{C}$ is $\mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$ then $\mathbf{C}^{+} = \mathbf{V}\Sigma^{-1}\mathbf{U}^\mathsf{T}$.
  - So, $\mathbf{C}\mathbf{C}^{+} = \mathbf{U}\mathbf{U}^\mathsf{T} = \mathbf{U}_{\mathrm{rank}(\mathbf{C})}\mathbf{U}^\mathsf{T}_{\mathrm{rank}(\mathbf{C})}$.

# Low rank approximation in small space

Problem with the previous theorem (substituting $\mathbf{C}$ for $\boldsymbol{A}$): calculating SVD of $\mathbf{C}$ is costly even if we use the small matrix $\mathbf{C}^{\mathsf{T}}\mathbf{C} \in \mathbb{R}^{s \times s}$.

- Idea: Use $\mathbf{C}\mathbf{C}^{+}$ instead of $\mathbf{U}_k\mathbf{U}_k^{\mathsf{T}}$.
  - $\mathbf{C}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.
  - If the SVD of $\mathbf{C}$ is $\mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$ then $\mathbf{C}^{+} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{\mathsf{T}}$.
  - So, $\mathbf{C}\mathbf{C}^{+} = \mathbf{U}\mathbf{U}^{\mathsf{T}} = \mathbf{U}_{\text{rank}(\mathbf{C})}\mathbf{U}^{\mathsf{T}}_{\text{rank}(\mathbf{C})}$.
  - But calculating the Moore-Penrose pseudoinverse is easier than the SVD. It reduces to inverse calculation which can be done in small space.

# Low rank approximation in small space cont.

**Theorem**

*Let the input matrix be* $\mathbf{M} \in \mathbb{R}^{m \times n}$. *Suppose that, for some* $\tilde{r} \in \mathbb{Z}^{+}$, $\tilde{r} \leqslant \mathrm{rank}(\mathbf{M})$ *and* $0 \leqslant \varepsilon \leqslant 0.9$, *it holds that* $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ *where* $\mathbf{M}_{\tilde{r}}$ *is the best* $\tilde{r}$*-rank approximation of* $\mathbf{M}$.

# Low rank approximation in small space cont.

## Theorem

*Let the input matrix be* $\mathbf{M} \in \mathbb{R}^{m \times n}$. *Suppose that, for some* $\tilde{r} \in \mathbb{Z}^+$, $\tilde{r} \leqslant \operatorname{rank}(\mathbf{M})$ *and* $0 \leqslant \varepsilon \leqslant 0.9$, *it holds that* $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ *where* $\mathbf{M}_{\tilde{r}}$ *is the best* $\tilde{r}$*-rank approximation of* $\mathbf{M}$. *Then, for every constant* $\delta > 0$, *there exists an* $O(\tilde{r} \log \tilde{r} + \log(mn))$*-space deterministic algorithm that outputs* $\mathbf{C}$ *and* $\mathbf{C}^+\mathbf{M}$

# Low rank approximation in small space cont.

**Theorem**

*Let the input matrix be $\mathbf{M} \in \mathbb{R}^{m \times n}$. Suppose that, for some $\tilde{r} \in \mathbb{Z}^+$, $\tilde{r} \leqslant \mathrm{rank}(\mathbf{M})$ and $0 \leqslant \varepsilon \leqslant 0.9$, it holds that $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ where $\mathbf{M}_{\tilde{r}}$ is the best $\tilde{r}$-rank approximation of $\mathbf{M}$. Then, for every constant $\delta > 0$, there exists an $O(\tilde{r} \log \tilde{r} + \log(mn))$-space deterministic algorithm that outputs $\mathbf{C}$ and $\mathbf{C}^+\mathbf{M}$ for which*

$$\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\right\|_F \leqslant (\varepsilon + \delta) \|\mathbf{M}\|_F$$

# Low rank approximation in small space cont.

## Theorem

*Let the input matrix be* $\mathbf{M} \in \mathbb{R}^{m \times n}$*. Suppose that, for some* $\tilde{r} \in \mathbb{Z}^{+}$*,* $\tilde{r} \leqslant \mathrm{rank}(\mathbf{M})$ *and* $0 \leqslant \varepsilon \leqslant 0.9$*, it holds that* $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ *where* $\mathbf{M}_{\tilde{r}}$ *is the best* $\tilde{r}$*-rank approximation of* $\mathbf{M}$*. Then, for every constant* $\delta > 0$*, there exists an* $O(\tilde{r} \log \tilde{r} + \log(mn))$*-space deterministic algorithm that outputs* $\mathbf{C}$ *and* $\mathbf{C}^{+}\mathbf{M}$ *for which*

$$\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^{+}\mathbf{M}\right\|_F \leqslant (\varepsilon + \delta) \|\mathbf{M}\|_F$$

*and the number of columns of* $\mathbf{C}$ *is* $O(\tilde{r})$*.*

# Low rank approximation in small space cont.

**Theorem**

*Let the input matrix be* $\mathbf{M} \in \mathbb{R}^{m \times n}$. *Suppose that, for some* $\tilde{r} \in \mathbb{Z}^{+}$, $\tilde{r} \leqslant \text{rank}(\mathbf{M})$ *and* $0 \leqslant \varepsilon \leqslant 0.9$, *it holds that* $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ *where* $\mathbf{M}_{\tilde{r}}$ *is the best* $\tilde{r}$*-rank approximation of* $\mathbf{M}$. *Then, for every constant* $\delta > 0$, *there exists an* $O(\tilde{r} \log \tilde{r} + \log(mn))$*-space deterministic algorithm that outputs* $\mathbf{C}$ *and* $\mathbf{C}^{+}\mathbf{M}$ *for which*

$$\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^{+}\mathbf{M}\right\|_F \leqslant (\varepsilon + \delta) \|\mathbf{M}\|_F$$

*and the number of columns of* $\mathbf{C}$ *is* $O(\tilde{r})$. *If* $\tilde{r} = O\left(\frac{\log(mn)}{\log(\log(mn))}\right)$ *then the space bound of the algorithm is* $O(\log(mn))$.

# Low rank approximation in small space cont.

**Theorem**

*Let the input matrix be $\mathbf{M} \in \mathbb{R}^{m \times n}$. Suppose that, for some $\tilde{r} \in \mathbb{Z}^+$, $\tilde{r} \leqslant \text{rank}(\mathbf{M})$ and $0 \leqslant \varepsilon \leqslant 0.9$, it holds that $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ where $\mathbf{M}_{\tilde{r}}$ is the best $\tilde{r}$-rank approximation of $\mathbf{M}$. Then, for every constant $\delta > 0$, there exists an $O(\tilde{r} \log \tilde{r} + \log(mn))$-space deterministic algorithm that outputs $\mathbf{C}$ and $\mathbf{C}^+\mathbf{M}$ for which*

$$\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\right\|_F \leqslant (\varepsilon + \delta) \|\mathbf{M}\|_F$$

*and the number of columns of $\mathbf{C}$ is $O(\tilde{r})$. If $\tilde{r} = O\left(\frac{\log(mn)}{\log(\log(mn))}\right)$ then the space bound of the algorithm is $O(\log(mn))$.*

Intuitively, $\mathbf{C}$ must "cover" most of the space that is spanned by the singular vectors corresponding to the top $\tilde{r}$ singular values of $\mathbf{M}$.

# Low rank approximation in small space cont.

**Theorem**

*Let the input matrix be $\mathbf{M} \in \mathbb{R}^{m \times n}$. Suppose that, for some $\tilde{r} \in \mathbb{Z}^+$, $\tilde{r} \leqslant \operatorname{rank}(\mathbf{M})$ and $0 \leqslant \varepsilon \leqslant 0.9$, it holds that $\|\mathbf{M} - \mathbf{M}_{\tilde{r}}\|_F \leqslant \varepsilon \|\mathbf{M}\|_F$ where $\mathbf{M}_{\tilde{r}}$ is the best $\tilde{r}$-rank approximation of $\mathbf{M}$. Then, for every constant $\delta > 0$, there exists an $O(\tilde{r}\log\tilde{r} + \log(mn))$-space deterministic algorithm that outputs $\mathbf{C}$ and $\mathbf{C}^+\mathbf{M}$ for which*

$$\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^+\mathbf{M}\right\|_F \leqslant (\varepsilon + \delta)\,\|\mathbf{M}\|_F$$

*and the number of columns of $\mathbf{C}$ is $O(\tilde{r})$. If $\tilde{r} = O\left(\frac{\log(mn)}{\log(\log(mn))}\right)$ then the space bound of the algorithm is $O(\log(mn))$.*

Intuitively, $\mathbf{C}$ must “cover” most of the space that is spanned by the singular vectors corresponding to the top $\tilde{r}$ singular values of $\mathbf{M}$. Then $\mathbf{C}\mathbf{C}^+$ is a projector that projects to this “covered” subspace.

# Note on the space bound

- If $\tilde{r} \ll n$ then $O(\tilde{r} \log \tilde{r} + \log(mn))$ space is less than
  - storing $\tilde{r}$ column indices, which would require $\tilde{r} \cdot \log n$ space,

# Note on the space bound

- If $\tilde{r} \ll n$ then $O(\tilde{r} \log \tilde{r} + \log(mn))$ space is less than
  - storing $\tilde{r}$ column indices, which would require $\tilde{r} \cdot \log n$ space,
  - storing the small $\mathbf{C}^\top \mathbf{C} \in \mathbb{R}^{s \times s}$ matrix, which would require space $s^2 = \Omega(\tilde{r}^2)$.

# Note on the space bound

- If $\tilde{r} \ll n$ then $O(\tilde{r} \log \tilde{r} + \log(mn))$ space is less than
  - storing $\tilde{r}$ column indices, which would require $\tilde{r} \cdot \log n$ space,
  - storing the small $\mathbf{C}^\mathsf{T}\mathbf{C} \in \mathbb{R}^{s \times s}$ matrix, which would require space $s^2 = \Omega(\tilde{r}^2)$.
- We can't even calculate the rank of $\mathbf{M}$ in this space bound, even if $\varepsilon = 0$.

# Note on the space bound

- If $\tilde{r} \ll n$ then $O(\tilde{r} \log \tilde{r} + \log(mn))$ space is less than
  - storing $\tilde{r}$ column indices, which would require $\tilde{r} \cdot \log n$ space,
  - storing the small $\mathbf{C}^\mathsf{T}\mathbf{C} \in \mathbb{R}^{s \times s}$ matrix, which would require space $s^2 = \Omega(\tilde{r}^2)$.
- We can't even calculate the rank of $\mathbf{M}$ in this space bound, even if $\varepsilon = 0$.
- Matrix multiplication can be done in this space bound so we can also output $\mathbf{C}\mathbf{C}^+\mathbf{M}$ and $\mathbf{C}^+$.

# Moore-Penrose pseudoinverse of $\boldsymbol{M}$

We can calculate the Moore-Penrose pseudoinverse of $\boldsymbol{M}$ if $\boldsymbol{M}$ is low rank.

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

**Theorem**

*Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be given as input, where the rank of $\mathbf{M}$ is $r$ and its condition number is constant.*

SZTAKI

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

**Theorem**

*Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be given as input, where the rank of $\mathbf{M}$ is $r$ and its condition number is constant. There exists a deterministic algorithm that outputs $\mathbf{M}^{+}$ using space $O\left(r^2 \log r + \log(mn)\right)$.*

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

**Theorem**

*Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be given as input, where the rank of $\mathbf{M}$ is $r$ and its condition number is constant. There exists a deterministic algorithm that outputs $\mathbf{M}^+$ using space $O\left(r^2 \log r + \log(mn)\right)$.*

*If $r = O\left(\frac{\sqrt{\log(mn)}}{\log(\log(mn))}\right)$ then the space bound becomes $O(\log(mn))$.*

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

**Theorem**

*Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be given as input, where the rank of $\mathbf{M}$ is $r$ and its condition number is constant. There exists a deterministic algorithm that outputs $\mathbf{M}^{+}$ using space $O\left(r^2 \log r + \log(mn)\right)$.*

*If $r = O\left(\frac{\sqrt{\log(mn)}}{\log(\log(mn))}\right)$ then the space bound becomes $O(\log(mn))$.*

- Bound gets worse: $\tilde{r} \to r^2$.

SZTAKI

# Moore-Penrose pseudoinverse of $\mathbf{M}$

We can calculate the Moore-Penrose pseudoinverse of $\mathbf{M}$ if $\mathbf{M}$ is low rank. Approximately low rank is not enough, because we have to invert all non-zero singular values.

**Theorem**

*Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be given as input, where the rank of $\mathbf{M}$ is $r$ and its condition number is constant. There exists a deterministic algorithm that outputs $\mathbf{M}^+$ using space $O\left(r^2 \log r + \log(mn)\right)$.*

*If $r = O\left(\frac{\sqrt{\log(mn)}}{\log(\log(mn))}\right)$ then the space bound becomes $O(\log(mn))$.*

- Bound gets worse: $\tilde{r} \to r^2$. This is because the sampled matrix must “cover” the whole space spanned by the singular vectors of $\mathbf{M}$.

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^{\mathsf{T}}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For $\mathbf{M} \in \mathbb{R}^{m \times n}$, let $r = \operatorname{rank}(\mathbf{M})$ and suppose its condition number is constant.*

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^{\mathsf{T}}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For $\mathbf{M} \in \mathbb{R}^{m \times n}$, let $r = \mathrm{rank}(\mathbf{M})$ and suppose its condition number is constant. For every constant $\delta > 0$, there exists a deterministic alg. that outputs $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{W} \in \mathbb{R}^{n \times r}$*

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For $\mathbf{M} \in \mathbb{R}^{m\times n}$, let $r = \mathrm{rank}(\mathbf{M})$ and suppose its condition number is constant. For every constant $\delta > 0$, there exists a deterministic alg. that outputs $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{W} \in \mathbb{R}^{n\times r}$ for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M},$$

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

## Theorem

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$*, let* $r = \operatorname{rank}(\mathbf{M})$ *and suppose its condition number is constant. For every constant* $\delta > 0$*, there exists a deterministic alg. that outputs* $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{W} \in \mathbb{R}^{n \times r}$ *for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

SZTAKI

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *let* $r = \operatorname{rank}(\mathbf{M})$ *and suppose its condition number is constant. For every constant* $\delta > 0$, *there exists a deterministic alg. that outputs* $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{W} \in \mathbb{R}^{n \times r}$ *for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

$$\left\|\tilde{\Sigma}^2 - \Sigma^2\right\|_\mathsf{F} = \frac{\delta \left\|\mathbf{M}\right\|_\mathsf{F}^2}{r},$$

$\Sigma = \operatorname{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$.

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For* $\mathbf{M} \in \mathbb{R}^{m\times n}$*, let* $r = \mathrm{rank}(\mathbf{M})$ *and suppose its condition number is constant. For every constant* $\delta > 0$*, there exists a deterministic alg. that outputs* $\mathbf{U} \in \mathbb{R}^{m\times r}$*,* $\tilde{\Sigma} \in \mathbb{R}^{r\times r}$*,* $\mathbf{W} \in \mathbb{R}^{n\times r}$ *for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

$$\left\|\tilde{\Sigma}^2 - \Sigma^2\right\|_\mathsf{F} = \frac{\delta\,\|\mathbf{M}\|_\mathsf{F}^2}{r}, \qquad \left\|\mathbf{W}^\mathsf{T}\mathbf{W} - \mathbb{1}\right\|_\mathsf{F} < \delta.$$

$\Sigma = \mathrm{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$.

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

## Theorem

*For* $\mathbf{M} \in \mathbb{R}^{m\times n}$, *let* $r = \mathrm{rank}(\mathbf{M})$ *and suppose its condition number is constant. For every constant* $\delta > 0$, *there exists a deterministic alg. that outputs* $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{W} \in \mathbb{R}^{n\times r}$ *for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

$$\left\|\tilde{\Sigma}^2 - \Sigma^2\right\|_\mathsf{F} = \frac{\delta\,\|\mathbf{M}\|_\mathsf{F}^2}{r}, \qquad \left\|\mathbf{W}^\mathsf{T}\mathbf{W} - \mathbb{1}\right\|_\mathsf{F} < \delta.$$

$\Sigma = \mathrm{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$. *The alg. uses space* $O\left(r^4 + \log(mn)\right)$.

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For $\mathbf{M} \in \mathbb{R}^{m\times n}$, let $r = \operatorname{rank}(\mathbf{M})$ and suppose its condition number is constant. For every constant $\delta > 0$, there exists a deterministic alg. that outputs $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{W} \in \mathbb{R}^{n\times r}$ for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

$$\left\|\tilde{\Sigma}^2 - \Sigma^2\right\|_\mathsf{F} = \frac{\delta\,\|\mathbf{M}\|_\mathsf{F}^2}{r}, \qquad \left\|\mathbf{W}^\mathsf{T}\mathbf{W} - \mathbb{1}\right\|_\mathsf{F} < \delta.$$

$\Sigma = \operatorname{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$. *The alg. uses space* $O\left(r^4 + \log(mn)\right)$. *If* $r = O\left(\log^{1/4}(mn)\right)$ *then the space bound becomes* $O(\log(mn))$.

SZTAKI

# Singular value decomposition of $\mathbf{M}$

If we can store $\mathbf{C}^\mathsf{T}\mathbf{C}$ then we can calculate an approximate SVD of $\mathbf{M}$.

**Theorem**

*For* $\mathbf{M} \in \mathbb{R}^{m \times n}$, *let* $r = \operatorname{rank}(\mathbf{M})$ *and suppose its condition number is constant. For every constant* $\delta > 0$, *there exists a deterministic alg. that outputs* $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{W} \in \mathbb{R}^{n \times r}$ *for which*

$$\mathbf{U}\tilde{\Sigma}\mathbf{W}^\mathsf{T} = \mathbf{M}, \qquad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbb{1},$$

$$\left\|\tilde{\Sigma}^2 - \Sigma^2\right\|_F = \frac{\delta \left\|\mathbf{M}\right\|_F^2}{r}, \qquad \left\|\mathbf{W}^\mathsf{T}\mathbf{W} - \mathbb{1}\right\|_F < \delta.$$

$\Sigma = \operatorname{diag}(\sigma_1(\mathbf{M}), \sigma_2(\mathbf{M}), \ldots, \sigma_r(\mathbf{M}))$. *The alg. uses space* $O\left(r^4 + \log(mn)\right)$. *If* $r = O\left(\log^{1/4}(mn)\right)$ *then the space bound becomes* $O(\log(mn))$.

The bound gets worse: $r^2 \to r^4$.

# Summary

- We saw a sampling method to handle low rank matrices.

## Summary

- We saw a sampling method to handle low rank matrices.
- Reduced the randomness and derandomized the method.

# Summary

- We saw a sampling method to handle low rank matrices.
- Reduced the randomness and derandomized the method.
- Three applications:
    - Low rank approximation

# Summary

- We saw a sampling method to handle low rank matrices.
- Reduced the randomness and derandomized the method.
- Three applications:
  - Low rank approximation
  - Calculating the Moore-Penrose pseudoinverse

# Summary

- We saw a sampling method to handle low rank matrices.
- Reduced the randomness and derandomized the method.
- Three applications:
  - Low rank approximation
  - Calculating the Moore-Penrose pseudoinverse
  - Calculating the SVD

# Summary

- We saw a sampling method to handle low rank matrices.
- Reduced the randomness and derandomized the method.
- Three applications:
    - Low rank approximation
    - Calculating the Moore-Penrose pseudoinverse
    - Calculating the SVD
- Open question:
    - What other interesting matrix properties can we use it for?

# Thank you for your attention!

Got comments, questions, ideas?

Email me at peresz@sztaki.hu