4D – Eigenanalysis & Dimensionality Reduction: SVD Applications

Derek Nowrouzezahrai
derek@cim.mcgill.ca

Singular Value Decomposition (SVD)

LU $\quad \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} L \end{bmatrix} \begin{bmatrix} U \end{bmatrix}$

QR $\quad \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} Q \end{bmatrix} \begin{bmatrix} R \end{bmatrix}$

SVD $\quad \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \begin{bmatrix} \Sigma \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix}$

Singular Value Decomposition (SVD)

LU and QR decompositions serve primarily as tools to solve linear systems

While the SVD is another matrix factorization that can applied in this context, it is much more versatile…

SVD – Outer Product Interpretation

We've discussed the inner product of two vectors

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v} = \textstyle\sum_i u_i v_i$$

but the *outer product* is not as commonly know:

$$\mathbf{u}\mathbf{v}^T = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \dots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \dots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \dots & u_m v_n \end{bmatrix}$$

- this <u>matrix</u> is also sometimes referred to as a *rank-1 matrix*

SVD – Outer Product Interpretation

We can reinterpret the SVD of a matrix

$$\begin{bmatrix}\mathbf{A}\end{bmatrix} = \begin{bmatrix}\mathbf{U}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}\end{bmatrix}\begin{bmatrix}\mathbf{V}^T\end{bmatrix}$$

(*diagram is not to scale*)

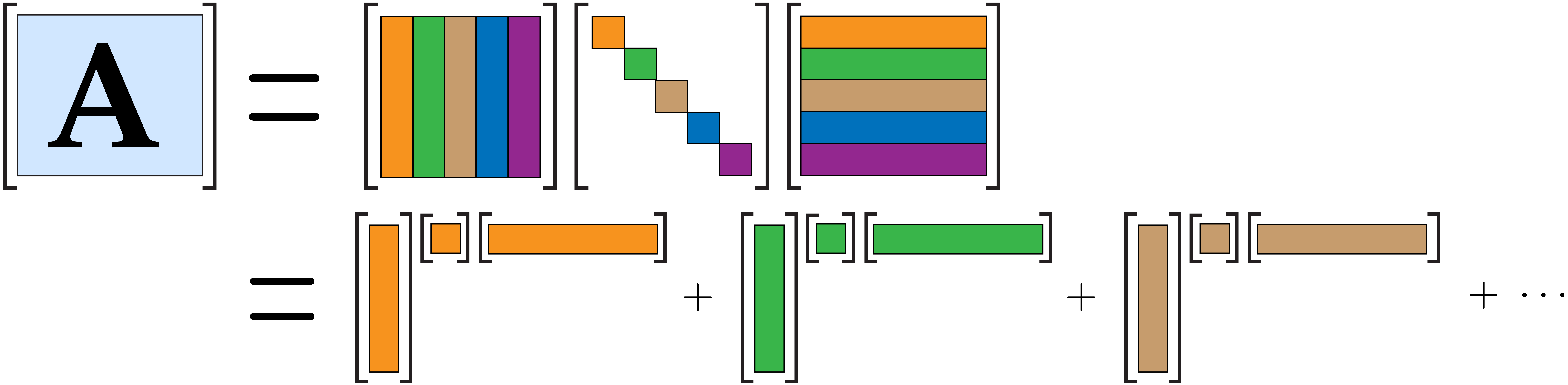
as a weighted sum of outer products, as

$$\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

SVD – Outer Product Interpretation

$$\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

This relationship can be visualized diagrammatically as



Computing the SVD

Computing the SVD

Much like the eigendecomposition, I will overview the general strategy before providing one example

- similarly, if you plan on working on areas of fundamental SVD research, you'll have to familiarize yourself with modern SVD computation strategies
- if you just want to use the SVD, you can rely on libraries

Computing the SVD

1. Start by computing $\mathbf{V}$ as the eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$
   - using your favourite eigendecomposition algorithm
2. From $\mathbf{A} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^{\mathrm{T}}$, we have $\mathbf{A}\,\mathbf{V} = \mathbf{U}\,\boldsymbol{\Sigma}$
   - so the columns of $\mathbf{U}$ corresponding to the nonzero singular values in $\boldsymbol{\Sigma}$ can be computed as normalized columns of $\mathbf{A}\,\mathbf{V}$
3. Compute the remaining columns of $\mathbf{U}$ by solving $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{u}_i = \mathbf{0}$
   - using your favourite linear systems solver, here

The Singular Value Decomposition

- 🙂 Can be computed for **any** matrix (non-square, non-symmetric, singular, …)
- 🙂 Computation is stable, i.e., does not require pivoting
- 🙂 Involves only orthogonal and diagonal matrices
- 🙂 **Impeccable** numerical properties
- 🙂 No complex arithmetic necessary
- 🙁 Expensive to compute (roughly 5 – 10x the cost of LU)

SVD Application 1: Linear Systems

SVD Applications – Linear Systems

As with the LU and QR decompositions, we can use the SVD to solve linear systems of equations

With a square, invertible matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, we arrive at a trivial solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ as $\mathbf{x} = \mathbf{V}\,\mathbf{\Sigma}^{-1}\,\mathbf{U}^{\mathrm{T}}\,\mathbf{b}$

SVD Applications – Linear Systems

With $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $m \neq n$ we solve $\mathbf{A}\mathbf{x} \approx \mathbf{b}$

- so far in the course, we've treated the $m > n$ overdetermined setting, solving for an $\mathbf{x}$ that minimizes the squared residual*
- in underdetermined settings, where $m < n$, an *infinite* number of solution vectors $\mathbf{x}$ satisfy $\mathbf{A}\mathbf{x} = \mathbf{b}$, and so we typically enforce an additional constraint on the norm of the solution

Solving the normal equations $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$ subject to minimizing $\|\mathbf{x}\|^2$ covers all three cases

SVD Applications – Linear Systems

With $\mathbf{A} = \mathbf{U\Sigma V}^{\mathrm{T}}$ we rewrite normal equations $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$ as

$$(\mathbf{U\Sigma V}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{U\Sigma V}^{\mathrm{T}})\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$(\mathbf{V\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}})(\mathbf{U\Sigma V}^{\mathrm{T}})\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{V\Sigma}^{\mathrm{T}}\mathbf{\Sigma V}^{\mathrm{T}}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{V\Sigma}^{\mathrm{T}}\mathbf{\Sigma V}^{\mathrm{T}}\mathbf{x} = (\mathbf{U\Sigma V}^{\mathrm{T}})^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{V\Sigma}^{\mathrm{T}}\mathbf{\Sigma V}^{\mathrm{T}}\mathbf{x} = \mathbf{V\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma V}^{\mathrm{T}}\mathbf{x} = \mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{b}$$

- setting $\mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{d} = \mathbf{U}^{\mathrm{T}}\mathbf{b}$ we seek solutions to $\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma y} = \mathbf{\Sigma}^{\mathrm{T}}\mathbf{d}$ that minimize $\|\mathbf{y}\|^2 = \|\mathbf{x}\|^2$

SVD Applications – Linear Systems

With $\mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{d} = \mathbf{U}^{\mathrm{T}}\,\mathbf{b}$, and exploiting the diagonal structure of $\boldsymbol{\Sigma}$ we note that:

- solutions to $\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{y} = \boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{d}$ satisfy $\sigma_i^2 y_i = \sigma_i d_i$ for all $\sigma_i \neq 0$
    - so, $y_i = d_i/\sigma_i$ for all $\sigma_i \neq 0$, and
- when $\sigma_i = 0$, there are no constraints on y_i and so we might as well choose $y_i = 0$ so as to minimize $\|\mathbf{y}\|^2$

Piecing things together, we write the solution as $\mathbf{y} = \boldsymbol{\Sigma}^{+}\,\mathbf{d}$, where $\boldsymbol{\Sigma}^{+} \in \mathbb{R}^{n\times m}$ has elements $(\boldsymbol{\Sigma}^{+})_{ij} = 1/\sigma_i$ for $i = j$ and $\sigma_i \neq 0$, and 0 otherwise

SVD Applications – Pseudoinverse

Expanding our solution $\mathbf{y} = \boldsymbol{\Sigma}^{+}\,\mathbf{d}$ with $\mathbf{d} = \mathbf{U}^{\mathrm{T}}\,\mathbf{b}$ back out in terms of the original problem statement, we have

$$\mathbf{x} = \mathbf{V}\,\boldsymbol{\Sigma}^{+}\,\mathbf{U}^{\mathrm{T}}\,\mathbf{b}$$

and we define the **pseudoinverse** of $\mathbf{A} \in \mathbb{R}^{m\times n}$ as $\mathbf{A}^{+} = (\mathbf{V}\,\boldsymbol{\Sigma}^{+}\,\mathbf{U}^{\mathrm{T}}) \in \mathbb{R}^{n\times m}$ and this matrix respects the following:

- for square $\mathbf{A}$, the pseudoinverse is the inverse: $\mathbf{A}^{+} = \mathbf{A}^{-1}$
- for overdetermined systems $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m > n$, $\mathbf{A}^{+}\mathbf{b}$ yields the least squares solution to $\mathbf{A}\mathbf{x} \approx \mathbf{b}$
- for underdetermined systems ($m < n$), $\mathbf{A}^{+}\mathbf{b}$ gives the least squares solution to $\mathbf{A}\mathbf{x} \approx \mathbf{b}$ with the smallest norm $\|\mathbf{x}\|^2$

SVD Application 2: Norms & Condition

SVD Applications – Norms & Condition

Earlier, we discussed induced matrix norms according to:

- their formal definition, $\|\mathbf{A}\|_* = \max\{\|\mathbf{Ax}\|_*, \text{ where } \|\mathbf{x}\|_* = 1\}$,
- their geometric interpretation, and
- a few special-case examples of how to "compute" norms

Most recently, we saw that the critical points of $\|\mathbf{Ax}\|^2 = \mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{Ax}$ that satisfy $\|\mathbf{x}\| = 1$ also satisfy the eigenvalue relationship $\mathbf{A}^\mathrm{T}\mathbf{Ax} = [E(\mathbf{x})]\mathbf{x}$

From this, we can rewrite the induced L_2 matrix norm as:

$$\|\mathbf{A}\|_2 = \max\{\sqrt{\lambda} \text{ s.t. } \exists \mathbf{x} \in \mathrm{R}^n \text{ with } \mathbf{A}^T\mathbf{Ax} = \lambda\mathbf{x}\}$$

SVD Applications – Norms & Condition

$$\|\mathbf{A}\|_2 = \max\{\sqrt{\lambda} \text{ s.t. } \exists \mathbf{x} \in \mathrm{R}^n \text{ with } \mathbf{A}^T\mathbf{A}\mathbf{x} = \lambda\mathbf{x}\}$$

This is the first expression for the induced L_2 matrix norm that admits a *computable* algorithm:

- compute the eigendecomposition of $\mathbf{A}^\mathrm{T}\mathbf{A}$ and take the square root of the largest eigenvalue

Recall, however, that forming $\mathbf{A}^\mathrm{T}\mathbf{A}$ affects its conditioning, and so a more robust algorithm computes the SVD and takes $\|\mathbf{A}\|_2 = \max_i \sigma_i$

SVD Applications – Norms & Condition

Another benefit is that we can similarly show that $\|\mathbf{A}^{-1}\|_2 = \min_i \sigma_i$, so the **condition number** of A can be computed as

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$$
$$= \frac{\sigma_{\max}}{\sigma_{\min}}.$$

SVD Application 3: PCA

Using SVD for PCA

Given m data points in n-dimensional space, adjusted to have 0-mean, and arranged in a data matrix as

$$D = \begin{pmatrix} p_1^1 & p_1^2 & \cdots & p_1^m \\ p_2^1 & p_2^2 & \cdots & p_2^m \\ \vdots & \vdots & \vdots & \vdots \\ p_n^1 & p_n^2 & \cdots & p_n^m \end{pmatrix}$$ with an n x n covariance matrix is $C = DD^t$

- recalling that we related the eigendecomposition of an outer-product S.P.SD. matrix $\mathbf{D}^T\mathbf{D}$ to the SVD of $\mathbf{D} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^T$, we can similarly relate the SVD to $\mathbf{D}\mathbf{D}^T$ as $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T)^T = \mathbf{U}\,\boldsymbol{\Sigma}^2\,\mathbf{U}^T$
  - and so the rows of $\mathbf{U}$ are the PCA principal axes

SVD Application 4: Low-rank Approx.

SVD Applications – Low-rank Approx.

We can use the SVD to form a *reduced rank* approximation of a matrix

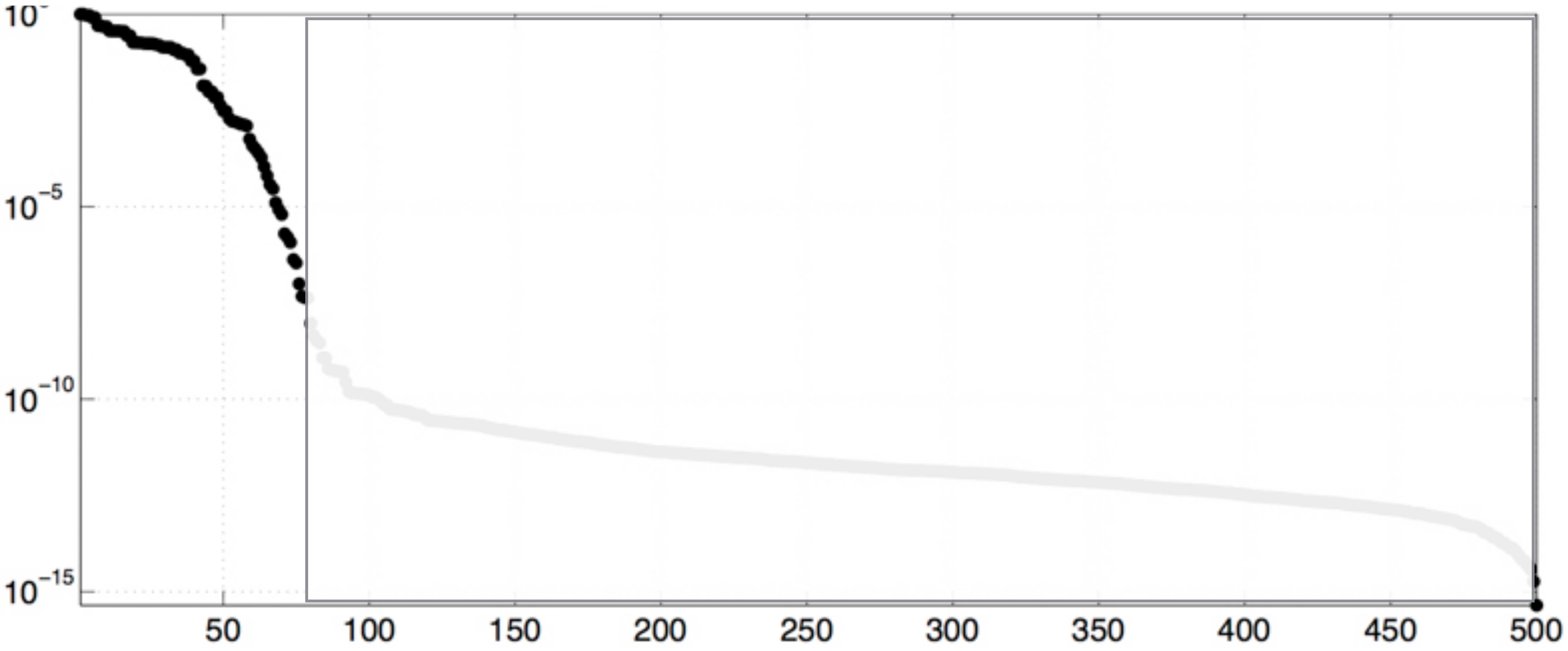
- starting from the outer product $\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ and only considering the first $k < n$ terms (i.e., singular values)

$$\mathbf{A} \approx \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{A}'$$

yields a *rank-k* approximation of $\mathbf{A}$ that is **provably optimal** in the L_2 (and Frobenius) sense: $\mathbf{A}'$ minimizes $\|\mathbf{A} - \mathbf{A}'\|_2$ among all rank-k matrices

SVD Applications – Low-rank Approx.

$$\mathbf{A} \approx \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{A}'$$

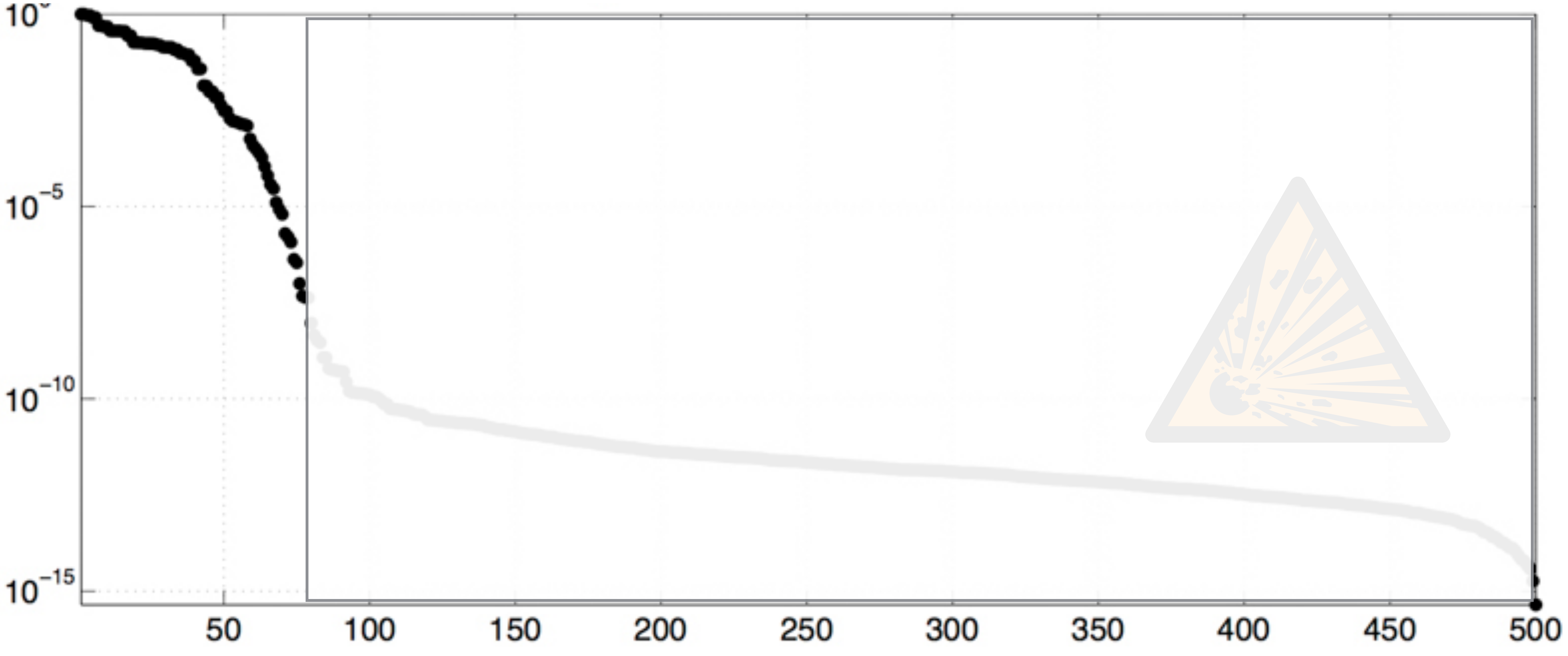


[Antonio Cosmin Ionita]

SVD Applications – Low-rank Approx.

Additionally, we can use this truncation to:

- more efficiently approximate products of the form $\mathbf{A}\mathbf{x}$
- approximate the inverse of poorly conditioned systems, recalling the definition of the pseudoinverse



SVD Example: Eigenfaces

Eigenfaces!

 …

$$\mathbf{A} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & \cdots & | \end{pmatrix}$$

[Sirovich and Kirby 1987]

Eigenfaces!

```
signature = [-6.85693, 23.7498, -11.4515,
-3.43352, 5.24749, -7.1615,  8.09015,
-9.7205, -0.660834, -2.4148, -10.3942,
3.33424,  2.94988, -2.75981, 3.02687,
-2.4499, -2.09885, -5.98832, -4.22564,
-0.65014, 2.20144, -5.43782, -9.61821,
-3.25227,   7.49413, -0.145002, 7.61483,
-0.696994, -3.7731, 3.23569, -1.78853,
0.0400116, -3.86804, -2.02456, 2.20949,
-1.86902,  1.23445, 0.140996, 0.698304,
-0.420466, 2.30691, 3.70434,  1.02417,
0.382809, 0.413049, -0.994902, 0.754145,
0.363418, -0.383865, 1.46379, 1.96381,
-2.90388, -2.33381, -0.438939, -0.30523,
-0.105925, 0.665962, -0.729409, -1.28977,
0.150497, 0.645343, 0.30724, -1.04942,
1.0462, -0.60808, 0.333288, 1.09659,
-1.38876, 0.33875, 0.278604, 1.0632,
-0.0446148, 0.24526, -0.283482, -0.236843,
0.312122]
```

[Sirovich and Kirby 1987]

Data-driven Norms

$\Big\|$  $-$  $\Big\|_? = d_0$

$\Big\|$  $-$  $\Big\|_? = d_1$

Want norm such that $d_0 << d_1$

[Sirovich and Kirby 1987]