UPPSALA
UNIVERSITET

Department of Information Technology

Scientific Computing for Data Analysis

Davoud Mirzaei

Lecture 9: Some applications of SVD

Agenda

- Least squares solution via SVD
- Low-rank approximation via SVD
- Principal component analysis (PCA)
- Mini-project 2

Least squares solution via SVD

$$\text{Given } A \in \mathbb{R}^{m\times n},\ m \geqslant n: \quad \min_{\boldsymbol{x}} \|A\boldsymbol{x} - \boldsymbol{b}\|_2$$

Case 1: If A is full rank ($\operatorname{rank}(A) = n$) then

$$A = U\Sigma V^T = [U_1 \;\; U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T, \quad \Sigma_1^{-1} \text{ exists}$$

By change of variables $\boldsymbol{y} = V^T\boldsymbol{x}$ we have

$$\begin{aligned} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 &= \|U\Sigma V^T\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|\Sigma V^T\boldsymbol{x} - U^T\boldsymbol{b}\|_2^2 = \left\| \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} U_1^T\boldsymbol{b} \\ U_2^T\boldsymbol{b} \end{bmatrix} \right\|_2^2 \\ &= \|\Sigma_1\boldsymbol{y} - U_1^T\boldsymbol{b}\|_2^2 + \|U_2^T\boldsymbol{b}\|_2^2. \end{aligned}$$

The right hand side is minimized if $\Sigma_1\boldsymbol{y} = U_1^T\boldsymbol{b}$ or $\boldsymbol{y} = \Sigma_1^{-1}U_1^T\boldsymbol{b}$. (since the term $\|U_2^T\boldsymbol{b}\|_2^2$ is independent of $\boldsymbol{y}$ and thus $\boldsymbol{x}$). Since $\boldsymbol{y} = V^T\boldsymbol{x}$ we can write

$$\boldsymbol{x} = V\Sigma_1^{-1}U_1^T\boldsymbol{b} = A^+\boldsymbol{b} \quad \text{(pseudoinverse)}$$

The residual is

$$residual = \|U_2^T\boldsymbol{b}\|_2$$

Least squares solution via SVD

Case 2: A is rank-deficient, $\text{rank}(A) = r < n$

$$A = U\Sigma V^T = [U_1 \;\; U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1 \;\; V_2]^T, \quad \Sigma_1^{-1} \text{ exists}$$

By change of variables $V^T\boldsymbol{x} =: \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}$, where $\boldsymbol{y}_1 \in \mathbb{R}^r$, we have

$$\begin{aligned} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 &= \|U\Sigma V^T\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|\Sigma V^T\boldsymbol{x} - U^T\boldsymbol{b}\|_2^2 \\ &= \left\| \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} - \begin{bmatrix} U_1^T\boldsymbol{b} \\ U_2^T\boldsymbol{b} \end{bmatrix} \right\|_2^2 \\ &= \|\Sigma_1\boldsymbol{y}_1 - U_1^T\boldsymbol{b}\|_2^2 + \|U_2^T\boldsymbol{b}\|_2^2. \end{aligned}$$

Regardless of $\boldsymbol{y}_2$, the value of $\|A\boldsymbol{x} - \boldsymbol{b}\|_2$ is minimized when $\Sigma_1\boldsymbol{y}_1 = U_1^T\boldsymbol{b}$, i.e.

$$\boldsymbol{y}_1 = \Sigma_1^{-1}U_1^T\boldsymbol{b}$$

Then we choose an arbitrary vector $\boldsymbol{y}_2$ and set $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}$ and

$$\boldsymbol{x} = V\boldsymbol{y}$$

.

Some notes:

- When A is rank deficient the problem has infinite number of solutions, as the vector $\boldsymbol{y}_2$ can be chosen arbitrarily
- The norm-minimal solution is the solution with minimum norm 2 among all solutions. The norm-minimal solution is obtained for particular case $\boldsymbol{y}_2 = 0$
- The residual of the least square problem is

$$residual = \|U_2^T \boldsymbol{b}\|_2$$

- It can be shown that the **norm-minimal solution** is

$$\boldsymbol{x} = V_1 \Sigma_1^{-1} U_1^T \boldsymbol{b} = A^+ \boldsymbol{b} \quad \text{(pseudoinverse)}$$

(Since $\boldsymbol{y}_1 = V_1^T \boldsymbol{x}$ and $\boldsymbol{y}_2 = V_2^T \boldsymbol{x}$)

Least squares solution via SVD

Steps of the algorithm for computing least square solution of $A\boldsymbol{x} = \boldsymbol{b}$:

1. Compute the SVD of A such that $A = U\Sigma V^T$
2. Determine r, the rank of A (number of nonzero singular values).
3. set U_1 the first r columns of U, V_1 the first r columns of V, and Σ_1 the leading $r \times r$ submatrix of Σ. Then

$$\boldsymbol{x} = V_1 \Sigma_1^{-1} U_1^T \boldsymbol{b}$$

4. $residual = \|U_2^T \boldsymbol{b}\|_2$.

- Note: When A is rank-deficient ($\text{rank}(A) = r < n$) the above procedure gives the norm-minimal solution.
- If the residual is not needed, the reduced SVD of A is enough.
- To compute the **numerical rank**, the values of σ_k less than $\epsilon_M \|A\|_\infty$, where ϵ_M is the machine epsilon, are accepted to be zero.

Example: The SVD factors of a matrix A are given by:

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Sigma = \begin{bmatrix} 2\sqrt{3} & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} V = \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 & 0 & \frac{-1}{\sqrt{3}} \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}$$

Solve the least square problem $\min \|A\boldsymbol{x} - \boldsymbol{b}\|_2$ for $\boldsymbol{b} = [1, 1, 1, 1, 1]^T$.

Solution: A is 5×4 and the rank of A is 2 (rank-deficient). We have infinite number of least squares solutions. The norm-minimal solution is $\boldsymbol{x} = V_1\Sigma_1^{-1}U_1^T\boldsymbol{b} = A^+\boldsymbol{b}$.In the last lecture we computed A^+, so we have

$$\boldsymbol{x} = A^+\boldsymbol{b} = \begin{bmatrix} \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Example (continue)

Example In the last example compute the residual.

Solution: The formula for residual is $residual = \|U_2^T \boldsymbol{b}\|_2$. Since $r = 2$ (rank), U_1 is the first 2 columns and U_2 the last 3 columns of U. So we have

$$U_2^T \boldsymbol{b} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

and

$$residual = \|U_2^T \boldsymbol{b}\|_2 = \sqrt{1+1+1} = \sqrt{3}.$$

Least squares solution via SVD

An exercise: Write a python code to obtain the least squares solution of the following system, and compute the residual.

$$Ax = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 2 & -1 \\ 1 & 1 & 0 \\ 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \cong \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \boldsymbol{b}$$

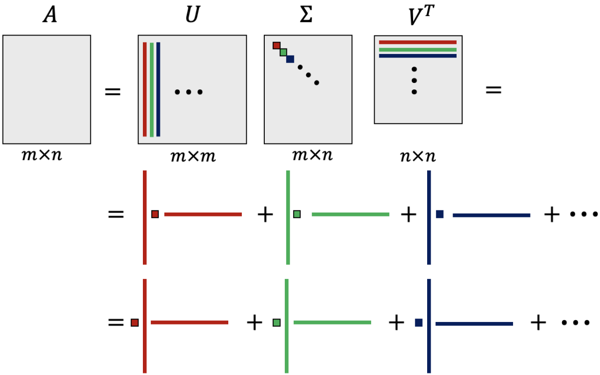
```
import numpy as np
A = np.array([[1,-1,2],[1,2,-1],[1,1,0],[1,3,-2]])
b = np.array([1,2,3,4])
U,S,Vt = np.linalg.svd(A)             # full SVD
eps = 2.2*10**-16;                    # machine epsilon
normA = np.linalg.norm(A,np.inf)      # norm infinity of A
r = np.size(np.where(S > eps*normA)) # numerical rank
U1 = U[:,:r]; V1 = Vt[:r,:].T; S1_inv = np.diag(1/S[:r])
Ap = V1@S1_inv@U1.T          # pseudo-inverse of A
x = Ap@b                     # norm minimal solution
print('norm minimal sol. x = ', x)
U2 = U[:,r+1:]               # U2 for residual
print('residual = ', np.linalg.norm(U2.T@b))
```

```
norm minimal sol. x =  [1.35238095 0.99047619 0.36190476]
residual =  0.9573111694631169
```

Low rank approximation via SVD

If $U = [\boldsymbol{u}_1 \ \ \boldsymbol{u}_2 \ \ \cdots \ \ \boldsymbol{u}_m]$ and $V = [\boldsymbol{v}_1 \ \ \boldsymbol{v}_2 \ \ \cdots \ \ \boldsymbol{v}_n]$ the we have

$$A = U\Sigma V^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_n \boldsymbol{u}_n \boldsymbol{v}_n^T$$

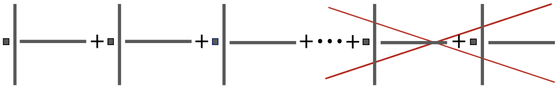


Matrices $\boldsymbol{u}_j \boldsymbol{v}_j^T$ are **rank 1 matrices** (outer product). So SVD writes A as a sum of rank 1 matrices. Since $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n$ matrix A is expressed as a list of its “ingredients”, ordered by “importance”

Low rank approximation via SVD

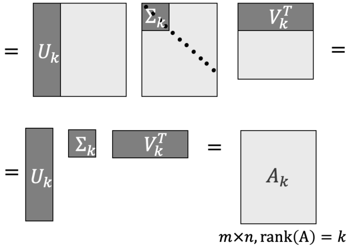
- Truncate the series at term k, for $k \leqslant n$, and define

$$A_k := \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$$



Keep the first k (most important) pieces, and throw away rest

- $A_k = U_k \Sigma_k V_k^T$, where Σ_k is derived from Σ by keeping the same diagonal entries, but setting the values of σ_{k+1} through σ_n to zero.



This idea is used for dimension reduction from n to k

Low rank approximation via SVD

- The rank of A_k is k ($\mathrm{rank}(A_k) = \mathrm{rank}(\Sigma_k) = k$) and we can show that A_k is the closet k rank matrix to A (in norm 2), i.e.

$$\|A - A_k\|_2 = \min_{rank(B)=k} \|A - B\|_2.$$

- $\|A - A_k\|_2 = \sigma_{k+1}$ because

$$\|A - A_k\|_2 = \|U\Sigma V^T - U\Sigma_k V^T\|_2 = \|\Sigma - \Sigma_k\|_2 = \sigma_{k+1}$$

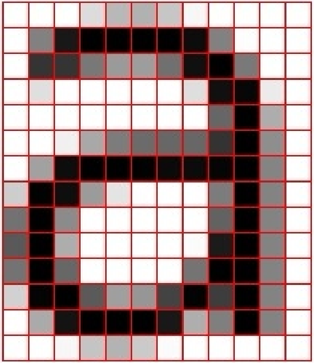
- Example: A matrix A has SVD with

$$\Sigma = \begin{bmatrix} 14.6 & 0 & 0 & 0 \\ 0 & 8.4 & 0 & 0 \\ 0 & 0 & 1.3 & 0 \\ 0 & 0 & 0 & 0.03 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

If we define $A_2 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T$, what is $\|A - A_2\|_2$?
Solution: It is $\sigma_3 = 1.3$

An application: image compression via SVD

- A gray image can be represented by an $m \times n$ matrix A whose (i,j)-th entry corresponds to the brightness of the pixel (i,j)
- The storage of this matrix requires mn locations



=

1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

An application: image compression via SVD

- The storage of this matrix requires mn locations
- The idea of image compression is to compress the image represented by a very large matrix to the one which corresponds to a lower-order approximation of A.
- SVD provides a simple way if one stores

$$\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T =: A_k$$

  instead
- The storage via SVD is $(m+n+1)k$ locations (the first k columns of U and V together with the first k singular values).
- This results a considerable savings when k is small.
- k should be also large enough to keep the quality of the image still acceptable

An application: image compression via SVD

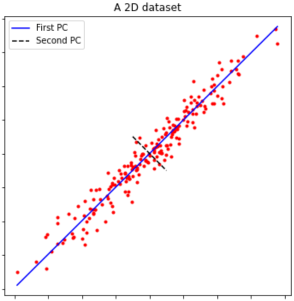
An Example:

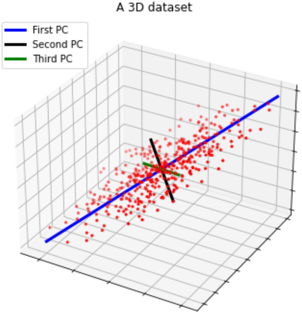
Another example in Lab 3.

An application: Principal Component Analysis (PCA) via SVD

Assume that we are given a dataset with observations on d variables (dimensions), for each of n entities or individuals. For example the weight, hight and age ($d = 3$ dimension) of $n = 1000$ individuals.

- What is the direction of the maximum variance in this data? (in which direction the data is widely spread?)
- What is the second most important direction in the data (and perpendicular to the first direction)?
- and so on ...





SVD answers the above questions:

- The data defines n vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n$ each in $\mathbb{R}^d$ or, equivalently, a $d \times n$ *data matrix*

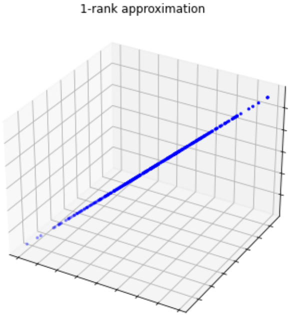
$$A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n]$$

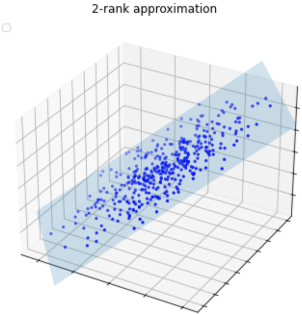
- Using SVD $A = U\Sigma V^T$, the columns $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_d$ are the directions of maximum variances in order of significance. These vectors are called principal components. $\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2$ are the corresponding variances.
- (**dimension reduction**) The rank k data matrix

$$A_k = U_k \Sigma_k V_k^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$$

  is a new k dimensional data that approximates the original data A (best k dimensional approximation)

Example from Lab 3: Consider the above 3D data. The rank 1 (1D, located on a line) and rank 2 (2D, located on a plane) approximations of the data are plotted:





$A_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ projects data to line $span\{\boldsymbol{u}_1\}$

$A_2 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T$ projects data to plane $span\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$

SVD: conclusion

- SVD is one the most important matrix decompositions with several applications
- Every matrix (either square or rectangular, either full rank or rank deficient) has SVD
- Can be used to solve the least squares problem with full rank or rank deficient coefficient matrix
- Can be used for computing pseudoinverse of a general matrix
- Applications to image processing
- Application to low rank approximation and dimension reduction (PCA)

Classification of handwritten digits

Download the training and test sets from Studium. Each digit (image) is a 28×28 matrix that is expressed by a vector of size 784.

1. For each digit, stack *training* images in front of each other and form a matrix A (so 10 big matrices)
2. Compute SVD of A and then use a few columns of U as an approximation space (10 SVD)
3. For each *test* digit, solve 10 least squares problems and compute the residuals
4. Classify the test digit in that class with the smallest residual