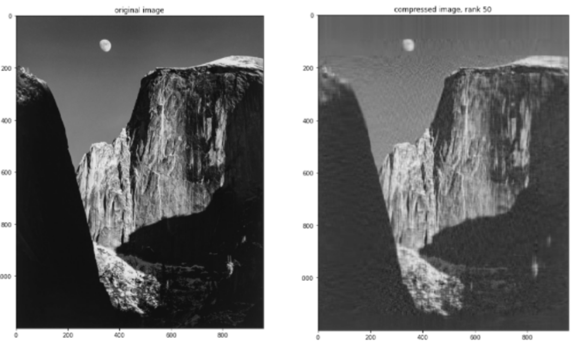
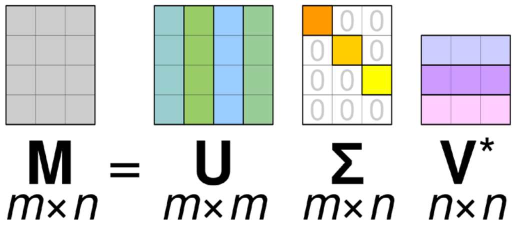


Lecture 12:
Singular Value Decomposition 1

CS 111: Intro to Computational Science
Spring 2023

Ziad Matni, Ph.D.
Dept. of Computer Science, UCSB



Images by Cmglee, https://commons.wikimedia.org/w/index.php?curid=67853297, Ansel Adams

Administrative

- New homework due Monday
- Lab tomorrow

- Quiz 3 grades now available on Canvas

RECALL: Eigendecomposition of a Matrix

- Consider the matrix Λ:
  n-by-n diagonal matrix with eigenvalues λ_j as elements

$$\Lambda = \begin{pmatrix} \lambda_0 & 0 & 0 & \dots \\ 0 & \lambda_1 & 0 & \dots \\ 0 & 0 & \lambda_2 & \dots \\ \dots & & & \end{pmatrix}$$

- ..and also consider the matrix **X**:

n-by-n set of corresponding eigenvectors $\mathbf{x}_j$ for each λ_j

$$X = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ X_0 & X_1 & X_2 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Leads to:

Eigen decomposition of a matrix: $\underline{\mathbf{A = X\Lambda X^{-1}}}$

Note the following again:

- Since **X** is columns, $\mathbf{X}^{-1}$ is "rotated-columns"
- Λ is a diagonal matrix

Matrix **Rank**

- Vector space dimension spanned by a matrix's columns
- The maximum number of *linearly independent columns* of a matrix.
  - In square matrices, that's also the max. number of linearly independent rows
- Examples – what ***rank*** do these matrices have?

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ -7 & 8 & 9 \end{bmatrix}$$

has a rank of **3** because it can represent 3 independent linear equations

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 6 & 9 \\ -2 & -4 & -6 \end{bmatrix}$$

has a rank of **1** because it can represent only 1 independent linear equation
(note how row2 = 3xrow1 and row3 = –2xrow1)

Matrix Rank

- The rank of any **m**x**n** matrix, **A**, has to be **≤** the smaller dimension:

  rank(A) ≤ min(m, n)

  - Rank(A) is always a positive number (*trivial exception: rank of a zero matrix is 0*)

- If **rank(A) = min(m,n)**, then **A** has a **full rank** (i.e. all columns are independent)
- Otherwise (i.e. rank < min size) **A** has a **deficient rank**

- A square matrix (where **m** = **n**), is invertible **ONLY** if **A** has a full rank

From a Computer's Perspective…

- Lower rank matrices can be "less useful" for certain computations (*example?!?!*)
- Example: you won't have a unique solution to **x** (in **Ax = b** problem) if **A** is nxn matrix with rank < n, because then it's not invertible

- HOWEVER, lower rank matrices can be represented in a computer while taking up *less storage space*!

- Matrices of Rank 1 happen to have their mathematical and computation advantages, as we shall see…

Matrix Rank Equivalencies

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ -1 & -2 & -3 & -4 \\ 10 & 20 & 30 & 40 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 10 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$$

This is called an <u>outer product</u>:
*Use **`np.outer()`***

Rank 1

Lower rank matrices can be represented with less storage space and can result in fewer calculations.

You can express a matrix of any rank as a sum of rank 1 matrices.

<u>Example:</u>

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ -1 & -2 & -3 & -4 \\ 6 & 7 & 8 & 9 \end{bmatrix}}_{\textbf{Rank 2}} = \underbrace{\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ -1 & -2 & -3 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\textbf{Rank 1}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 6 & 7 & 8 & 9 \end{bmatrix}}_{\textbf{Rank 1}}$$

Rank1 diagram from L.Serrano

Norm of a Matrix

A matrix norm $||\boldsymbol{A}||$ is any *mapping* from $\mathbb{R}^{nxn}$ to $\mathbb{R}$ with the following 3 properties:

1. $||\boldsymbol{A}|| > 0$ assuming $\boldsymbol{A} \neq 0$
2. $||\alpha \boldsymbol{A}|| = |\alpha|\ ||\boldsymbol{A}||$ for any $\alpha \in \mathbb{R}$
3. $||\boldsymbol{A} + \boldsymbol{B}|| \leq ||\boldsymbol{A}|| + ||\boldsymbol{B}||$

Norm of a **Matrix**

Common matrix norm types:

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|, \quad l_1 \text{ norm}$$

`npla.norm(A,1)`

$$\|A\|_2 = \max_{1\leq j\leq n} \sigma_{max}, \quad l_2 \text{ norm}$$

`npla.norm(A,2)`

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|, \quad l_\infty \text{ norm}$$

`npla.norm(A,np.inf)`

$$\boxed{||A||_F = [\textstyle\sum_{i,j} abs(a_{i,j})^2]^{1/2}}$$ Frobenius norm

```
npla.norm(A,'fro')
npla.norm(A)
```

Singular Value Decomposition

- SVD is a **factorization** of a matrix that **generalizes** the **eigendecomposition** of a *square matrix* to ***any*** **mxn matrix**:

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\ \underset{m\times n}{\boldsymbol{\Sigma}}\ \underset{n\times n}{\mathbf{V}^{T}}$$

- **U** an mxm *column orthogonal* matrix
- **V** an nxn *column orthogonal* matrix (so V^T is *row orthogonal*)
- **Σ** an mxn *diagonal* matrix whose elements $\boldsymbol{\sigma_i}$ are ordered:

$$\sigma_0 > \sigma_1 > \ldots\ \sigma_{min(m,n)-1} \geq 0$$

These σ_i are called the *singular values* of **A**

Singular Value Decomposition on Square Matrix

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

$$= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix}$$

$$= u_1 \sigma_1 v_1^T + u_2 \sigma_2 v_2^T + \dots + u_n \sigma_n v_n^T$$

$u_n \sigma_n v_n^T$ can be computed as:

```
u, sigma, vt = npla.svd(A)
sigma[n] * np.outer(u[:, n], vt[n, :])
```

Video Demo!

https://bit.ly/3tdNNsY   by: L. Serrano, 2020

- About 7 minutes long

- Describes how singular value decomposition (SVD) also describes the **linear transformation properties of a matrix**

Common Applications of SVD

- Principal Component Analysis (PCA)
  - Modelling data with minimal dimensions (similar to line-fitting)
- Discrete Optimization Problems
  - Dimension reduction
- Data File Compression
  - Image compression

SVD for Image Compression

- SVD is also a tool to help with image compression (think JPEGs…)
- More on this in the next lecture…
  - Along with a very cool demonstration…

Quick! To the Python-mobile!

Theorems Relating to SVD

1. The ***rank*** of **A** is the number of *nonzero* singular values (**number of σ_i**)

2. The ***2-norm*** $||\mathbf{A}||_2$ is equal to the largest singular value, i.e. σ_0

3. The ***2-norm condition number*** $\kappa_2(\mathbf{A})$ is equal to the ratio of the largest and smallest singular values. That is, $\kappa_2(A) = \sigma_0/\sigma_{\min(m,n)-1}$

4. The ***Frobenius norm*** $||\mathbf{A}||_F$ is equal to $(\Sigma_i\, \sigma_i^2)^{1/2}$

5. The ***determinant of a square matrix*** is the product of its singular values, $\Pi_i\sigma_i$

6. Matrix **A** is the sum of rank-1 matrices: $$A_k = \sum_{i=0}^{k-1} \sigma_i u_i v_i^T$$

Your TO DOs!

- Finish new assignment by Monday
- Lab tomorrow!

</LECTURE>