Matrix Analysis and Applications

Chapter 5: Singular Value Decomposition

Instructor: Kai Lu

(http://seit.sysu.edu.cn/teacher/1801)

School of Electronics and Information Technology
Sun Yat-sen University

November 24, 2020

Table of Contents

Table of Contents

Singular Value Decomposition (SVD)

Theorem 1 (Singular Value Decomposition)

Any $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ *can be factored as*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H, \tag{1}$$

where $\boldsymbol{U} \in \mathbb{C}^{m\times m}$ *and* $\boldsymbol{V} \in \mathbb{C}^{n\times n}$ *are unitary, and* $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ *has its elements given by*

$$[\boldsymbol{\Sigma}]_{ij} = \begin{cases} \sigma_i, & i = j \\ 0, & i \neq j \end{cases}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0,\ p \triangleq \min\{m, n\}$.

Singular Value Decomposition (SVD)

Theorem 1 (Singular Value Decomposition)

Any $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ *can be factored as*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H, \tag{1}$$

where $\boldsymbol{U} \in \mathbb{C}^{m \times m}$ *and* $\boldsymbol{V} \in \mathbb{C}^{n \times n}$ *are unitary, and* $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ *has its elements given by*

$$[\boldsymbol{\Sigma}]_{ij} = \begin{cases} \sigma_i, & i = j \\ 0, & i \neq j \end{cases}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0,\ p \triangleq \min\{m, n\}$.

Note:

- σ_i, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ are called the i^{th} **singular value**, **left singular vector**, and **right singular vector** of $\boldsymbol{A}$, respectively;
- if $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, then $\boldsymbol{U}$ and $\boldsymbol{V}$ may be taken to be real orthogonal matrices.

Representations of SVD

1) **Partitioned form**: let r be the number of nonzero singular values; i.e., $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$, $\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^H \\ \boldsymbol{V}_2^H \end{bmatrix}, \tag{2}$$

where $\tilde{\boldsymbol{\Sigma}} \triangleq \text{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$, $\boldsymbol{U}_1 \in \mathbb{C}^{m\times r}$, $\boldsymbol{U}_2 \in \mathbb{C}^{m\times(m-r)}$, $\boldsymbol{V}_1 \in \mathbb{C}^{n\times r}$, and $\boldsymbol{V}_2 \in \mathbb{C}^{n\times(n-r)}$.

Representations of SVD

1) **Partitioned form**: let r be the number of nonzero singular values; i.e., $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$, $\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^H \\ \boldsymbol{V}_2^H \end{bmatrix}, \tag{2}$$

where $\tilde{\boldsymbol{\Sigma}} \triangleq \text{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$, $\boldsymbol{U}_1 \in \mathbb{C}^{m\times r}$, $\boldsymbol{U}_2 \in \mathbb{C}^{m\times(m-r)}$, $\boldsymbol{V}_1 \in \mathbb{C}^{n\times r}$, and $\boldsymbol{V}_2 \in \mathbb{C}^{n\times(n-r)}$.

2) **Thin SVD form**:

$$\boldsymbol{A} = \boldsymbol{U}_1 \tilde{\boldsymbol{\Sigma}} \boldsymbol{V}_1^H. \tag{3}$$

Representations of SVD

1) **Partitioned form**: let r be the number of nonzero singular values; i.e., $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$, $\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^H \\ \boldsymbol{V}_2^H \end{bmatrix}, \tag{2}$$

where $\tilde{\boldsymbol{\Sigma}} \triangleq \mathrm{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$, $\boldsymbol{U}_1 \in \mathbb{C}^{m\times r}$, $\boldsymbol{U}_2 \in \mathbb{C}^{m\times(m-r)}$, $\boldsymbol{V}_1 \in \mathbb{C}^{n\times r}$, and $\boldsymbol{V}_2 \in \mathbb{C}^{n\times(n-r)}$.

2) **Thin SVD form**:

$$\boldsymbol{A} = \boldsymbol{U}_1 \tilde{\boldsymbol{\Sigma}} \boldsymbol{V}_1^H. \tag{3}$$

3) **Outer-product form**:

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H. \tag{4}$$

Some Observations

Suppose that $\boldsymbol{A}$ can be written in the SVD form $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$. Then,

$$\boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{U}\boldsymbol{D}_1\boldsymbol{U}^H, \quad \boldsymbol{D}_1 = \mathsf{diag}(\sigma_1^2, \cdots, \sigma_p^2, \underbrace{0, \cdots, 0}_{m-p \text{ zeros}}) \tag{5}$$

$$\boldsymbol{A}^H\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}_2\boldsymbol{V}^H, \quad \boldsymbol{D}_2 = \mathsf{diag}(\sigma_1^2, \cdots, \sigma_p^2, \underbrace{0, \cdots, 0}_{n-p \text{ zeros}}) \tag{6}$$

Some Observations

Suppose that $\boldsymbol{A}$ can be written in the SVD form $\boldsymbol{A} = \boldsymbol{U\Sigma V}^H$. Then,

$$\boldsymbol{AA}^H = \boldsymbol{UD}_1\boldsymbol{U}^H, \quad \boldsymbol{D}_1 = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2, \underbrace{0, \cdots, 0}_{m-p \text{ zeros}}) \tag{5}$$

$$\boldsymbol{A}^H\boldsymbol{A} = \boldsymbol{VD}_2\boldsymbol{V}^H, \quad \boldsymbol{D}_2 = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2, \underbrace{0, \cdots, 0}_{n-p \text{ zeros}}) \tag{6}$$

Observations: (ED vs. SVD)

- Eqs. (5) and (6) are the eigendecompositions of $\boldsymbol{AA}^H$ and $\boldsymbol{A}^H\boldsymbol{A}$, resp.;
- the left singular matrix $\boldsymbol{U}$ of $\boldsymbol{A}$ is the eigenvector matrix of $\boldsymbol{AA}^H$; the right singular matrix $\boldsymbol{V}$ of $\boldsymbol{A}$ is the eigenvector matrix of $\boldsymbol{A}^H\boldsymbol{A}$;
- the nonzero singular values $\sigma_1, \cdots, \sigma_r$ of $\boldsymbol{A}$, upon taking square, are the nonzero eigenvalues of both $\boldsymbol{AA}^H$ and $\boldsymbol{A}^H\boldsymbol{A}$.

Some Observations (cont'd)

Recall from Lecture 2 that the matrix 2-norm is defined as

$$\|\boldsymbol{A}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2. \tag{7}$$

Let $\sigma_i(\boldsymbol{A})$ denote the i^{th} singular value of $\boldsymbol{A}$.

Property 1

$$\|\boldsymbol{A}\|_2 = \sigma_1(\boldsymbol{A}),\ \|\boldsymbol{A}\|_F = \sqrt{\sigma_1^2(\boldsymbol{A}) + \cdots + \sigma_p^2(\boldsymbol{A})},\ p = \min\{m, n\}.$$

Some Observations (cont'd)

Recall from Lecture 2 that the matrix 2-norm is defined as

$$\|\boldsymbol{A}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2. \tag{7}$$

Let $\sigma_i(\boldsymbol{A})$ denote the i^{th} singular value of $\boldsymbol{A}$.

Property 1

$$\|\boldsymbol{A}\|_2 = \sigma_1(\boldsymbol{A}),\ \|\boldsymbol{A}\|_F = \sqrt{\sigma_1^2(\boldsymbol{A}) + \cdots + \sigma_p^2(\boldsymbol{A})},\ p = \min\{m, n\}.$$

Proof.

- Method 1: In lecture 2, we proved that $\|\boldsymbol{A}\|_2 = \sqrt{\lambda_{\max}(\boldsymbol{A}^H\boldsymbol{A})}$. Also, from the previous discussion, we have $\lambda_{\max}(\boldsymbol{A}^H\boldsymbol{A}) = \sigma_1^2(\boldsymbol{A})$.
- Method 2: Substitute the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$ into (7), and do the proof (details omitted).

□

Some Observations (cont'd)

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$. From the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, it is easy to verify that

1) $\boldsymbol{A}$ is nonsingular if and only if $\sigma_i > 0$ for all i.
2) The inverse of a nonsingular $\boldsymbol{A}$ is

$$\boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^H = \sum_{i=1}^{n} \frac{1}{\sigma_i} \boldsymbol{v}_i \boldsymbol{u}_i^H. \tag{8}$$

Some Observations (cont'd)

Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$. From the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, it is easy to verify that

1) $\boldsymbol{A}$ is nonsingular if and only if $\sigma_i > 0$ for all i.
2) The inverse of a nonsingular $\boldsymbol{A}$ is

$$\boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^H = \sum_{i=1}^{n} \frac{1}{\sigma_i}\boldsymbol{v}_i\boldsymbol{u}_i^H. \tag{8}$$

Remark 1 (More about matrix inverse)

- *The Sherman-Morrison formula:*

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}. \tag{9}$$

- *The Sherman-Morrison-Woodbury formula:*

$$\left(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^T\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{I} + \boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}^T\boldsymbol{A}^{-1}. \tag{10}$$

- *Perturbation and the inverse: If $\boldsymbol{A}$ is nonsingular and $r = \|\boldsymbol{A}^{-1}\boldsymbol{E}\|_p < 1$, then $\boldsymbol{A}+\boldsymbol{E}$ is nonsingular and*

$$\left\|(\boldsymbol{A}+\boldsymbol{E})^{-1} - \boldsymbol{A}^{-1}\right\|_p \le \frac{1}{1-r}\|\boldsymbol{E}\|_p \left\|\boldsymbol{A}^{-1}\right\|_p^2. \tag{11}$$

Insight of the SVD Proof

The matrix $\boldsymbol{A}\boldsymbol{A}^H$ is Hermitian and PSD. Thus, it admits an eigendecomposition

$$\boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H, \tag{12}$$

where $\boldsymbol{U}$ is the unitary eigenvector matrix, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_m) \succeq \mathbf{0}$.

Insight of the SVD Proof

The matrix $\boldsymbol{A}\boldsymbol{A}^H$ is Hermitian and PSD. Thus, it admits an eigendecomposition

$$\boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H, \tag{12}$$

where $\boldsymbol{U}$ is the unitary eigenvector matrix, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_m) \succeq \mathbf{0}$.

Suppose that $\lambda_1 \geq \cdots \geq \lambda_m > 0$. Let

$$\tilde{\boldsymbol{\Sigma}} \triangleq \boldsymbol{\Lambda}^{\frac{1}{2}} = \mathrm{diag}(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_m}) \text{ and } \boldsymbol{V}_1 \triangleq \boldsymbol{A}^H\boldsymbol{U}\tilde{\boldsymbol{\Sigma}}^{-1}. \tag{13}$$

It is easy to verify that

$$\boldsymbol{A} = \boldsymbol{U}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}_1^H \text{ and } \boldsymbol{V}_1^H\boldsymbol{V}_1 = \boldsymbol{I}. \tag{14}$$

Thus, a thin SVD has been established.

Insight of the SVD Proof

The matrix $\boldsymbol{A}\boldsymbol{A}^H$ is Hermitian and PSD. Thus, it admits an eigendecomposition

$$\boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H, \tag{12}$$

where $\boldsymbol{U}$ is the unitary eigenvector matrix, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \cdots, \lambda_m) \succeq \boldsymbol{0}$.

Suppose that $\lambda_1 \geq \cdots \geq \lambda_m > 0$. Let

$$\tilde{\boldsymbol{\Sigma}} \triangleq \boldsymbol{\Lambda}^{\frac{1}{2}} = \text{diag}(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_m}) \text{ and } \boldsymbol{V}_1 \triangleq \boldsymbol{A}^H\boldsymbol{U}\tilde{\boldsymbol{\Sigma}}^{-1}. \tag{13}$$

It is easy to verify that

$$\boldsymbol{A} = \boldsymbol{U}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}_1^H \text{ and } \boldsymbol{V}_1^H\boldsymbol{V}_1 = \boldsymbol{I}. \tag{14}$$

Thus, a thin SVD has been established.

Note: the full SVD proof requires

- finding $\boldsymbol{V}_2$ such that $\begin{bmatrix}\boldsymbol{V}_1 & \boldsymbol{V}_2\end{bmatrix}$ is unitary;
- covering instances where some λ_i's are zero.

Subspace and SVD

Property 2

By recalling (2), the following properties hold:

1) $\mathcal{R}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{U}_1)$;
2) $\mathcal{R}(\boldsymbol{A})_{\perp} = \mathcal{R}(\boldsymbol{U}_2)$;
3) $\mathrm{rank}(\boldsymbol{A}) = r$ (r *is the number of nonzero singular values*);
4) $\mathcal{R}(\boldsymbol{A}^H) = \mathcal{R}(\boldsymbol{V}_1)$;
5) $\mathcal{N}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{V}_2)$.

Subspace and SVD

Property 2

By recalling (2), the following properties hold:

1) $\mathcal{R}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{U}_1)$;
2) $\mathcal{R}(\boldsymbol{A})_\perp = \mathcal{R}(\boldsymbol{U}_2)$;
3) $\mathrm{rank}(\boldsymbol{A}) = r$ *(r is the number of nonzero singular values)*;
4) $\mathcal{R}(\boldsymbol{A}^H) = \mathcal{R}(\boldsymbol{V}_1)$;
5) $\mathcal{N}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{V}_2)$.

Note:

1) SVD provides a tool for finding bases of $\mathcal{R}(\boldsymbol{A}), \mathcal{R}(\boldsymbol{A})_\perp, \mathcal{R}(\boldsymbol{A}^H)$, and $\mathcal{N}(\boldsymbol{A})$;
2) Property 2 enables a simple proof of some basic matrix results, such as
    - $\mathrm{rank}(\boldsymbol{A}^T) = \mathrm{rank}(\boldsymbol{A})$;
    - $\dim \mathcal{N}(\boldsymbol{A}) = n - \mathrm{rank}(\boldsymbol{A})$.

Table of Contents

Implication to Linear Systems

Consider a linear system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{15}$$

where $\boldsymbol{x}$ is the system input; $\boldsymbol{y}$ the system output; $\boldsymbol{A}$ the system response. By recalling the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, the linear system input-output relationship can be decomposed as

$$\tilde{\boldsymbol{x}} = \boldsymbol{V}^H\boldsymbol{x}, \tag{16a}$$

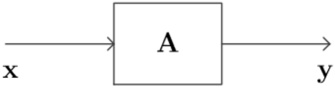
$$\tilde{\boldsymbol{y}} = \boldsymbol{\Sigma}\tilde{\boldsymbol{x}}, \tag{16b}$$

$$\boldsymbol{y} = \boldsymbol{U}\tilde{\boldsymbol{y}}. \tag{16c}$$

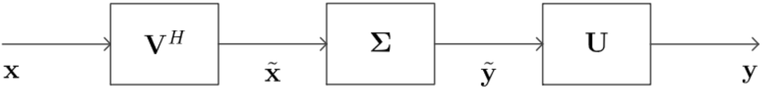
Implication: all linear systems work by performing three processes in cascade:

1) Eq. (16a) means to apply an orthogonal transformation (rotation and reflection) on the system input;
2) Eq. (16b) implies to apply non-negative scaling on the orthogonally transformed system input, and also add or remove some entries;
3) Eq. (16b) is another orthogonal transformation to yield the system output.

Implication to Linear Systems



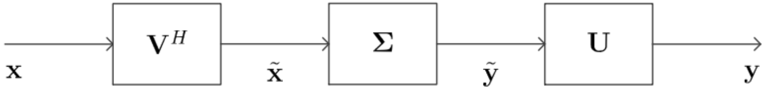
(a) linear system



(b) equivalent system

Implication to Linear Systems

(a) linear system



(b) equivalent system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{U}\underbrace{\boldsymbol{\Sigma}\underbrace{\boldsymbol{V}^H\boldsymbol{x}}_{\tilde{\boldsymbol{x}}}}_{\tilde{\boldsymbol{y}}}. \tag{17}$$

Systems of Linear Equations

Problem: Given $\boldsymbol{A} \in \mathbb{C}^{m \times n}$, $\boldsymbol{y} \in \mathbb{C}^{n}$, find an $\boldsymbol{x} \in \mathbb{C}^{m}$ such that

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{18}$$

1) it is well-known that if $\boldsymbol{A}$ is square and nonsingular, then (18) always has a solution and the solution is uniquely given by $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$;

2) how about other cases?

Systems of Linear Equations (cont'd)

Let $\tilde{\boldsymbol{x}} \triangleq \boldsymbol{V}^H \boldsymbol{x}$ and $\tilde{\boldsymbol{y}} \triangleq \boldsymbol{U}^H \boldsymbol{y}$. Partition

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_1^H \boldsymbol{x} \\ \boldsymbol{V}_2^H \boldsymbol{x} \end{bmatrix}, \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_1^H \boldsymbol{y} \\ \boldsymbol{U}_2^H \boldsymbol{y} \end{bmatrix}, \tag{19}$$

where $\tilde{\boldsymbol{x}}_1 \in \mathbb{C}^r$, $\tilde{\boldsymbol{x}}_2 \in \mathbb{C}^{n-r}$, $\tilde{\boldsymbol{y}}_1 \in \mathbb{C}^r$, and $\tilde{\boldsymbol{y}}_2 \in \mathbb{C}^{m-r}$.

Systems of Linear Equations (cont'd)

Let $\tilde{\boldsymbol{x}} \triangleq \boldsymbol{V}^H \boldsymbol{x}$ and $\tilde{\boldsymbol{y}} \triangleq \boldsymbol{U}^H \boldsymbol{y}$. Partition

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_1^H \boldsymbol{x} \\ \boldsymbol{V}_2^H \boldsymbol{x} \end{bmatrix}, \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_1^H \boldsymbol{y} \\ \boldsymbol{U}_2^H \boldsymbol{y} \end{bmatrix}, \tag{19}$$

where $\tilde{\boldsymbol{x}}_1 \in \mathbb{C}^r$, $\tilde{\boldsymbol{x}}_2 \in \mathbb{C}^{n-r}$, $\tilde{\boldsymbol{y}}_1 \in \mathbb{C}^r$, and $\tilde{\boldsymbol{y}}_2 \in \mathbb{C}^{m-r}$.

The linear system equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ can be equivalently transformed as

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \end{bmatrix}. \tag{20}$$

- The linear system problem reduces to that of the diagonal system above.

Systems of Linear Equations (cont'd)

Case 1: Suppose that $\boldsymbol{A}$ has full column rank; i.e., $m \geq n$, rank$(\boldsymbol{A}) = n$. (20) becomes

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} \\ \boldsymbol{0} \end{bmatrix} \tilde{\boldsymbol{x}} \Longleftrightarrow \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{y}}_1, \quad \tilde{\boldsymbol{y}}_2 = \boldsymbol{0}. \tag{21}$$

Note that $\tilde{\boldsymbol{y}}_2 = \boldsymbol{0}$ describes the condition for the linear system to have a solution.

Systems of Linear Equations (cont'd)

Case 1: Suppose that $\boldsymbol{A}$ has full column rank; i.e., $m \geq n$, rank$(\boldsymbol{A}) = n$. (20) becomes

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} \\ \boldsymbol{0} \end{bmatrix} \tilde{\boldsymbol{x}} \Longleftrightarrow \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{y}}_1, \quad \tilde{\boldsymbol{y}}_2 = \boldsymbol{0}. \tag{21}$$

Note that $\tilde{\boldsymbol{y}}_2 = \boldsymbol{0}$ describes the condition for the linear system to have a solution.

By transforming $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ back to $\boldsymbol{x}$ and $\boldsymbol{y}$, resp., the following results are concluded:

1) $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ has a solution if and only if $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{U}_1) = \mathcal{R}(\boldsymbol{A})$.

2) The solution, if exists, is uniquely given by

$$\boldsymbol{x} = \boldsymbol{V}\tilde{\boldsymbol{x}} \overset{(21)}{=\!=\!=} \boldsymbol{V}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{y}}_1 \overset{(19)}{=\!=\!=} \boldsymbol{V}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{U}_1^H \boldsymbol{y}. \tag{22}$$

Systems of Linear Equations (cont'd)

Case 2: Suppose that $\boldsymbol{A}$ has full row rank; i.e., $m \leq n$, rank$(\boldsymbol{A}) = m$. (20) becomes

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \end{bmatrix} \Longleftrightarrow \tilde{\boldsymbol{x}}_1 = \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{y}}, \quad \tilde{\boldsymbol{x}}_2 \text{ can be arbitrary.} \tag{23}$$

Systems of Linear Equations (cont'd)

Case 2: Suppose that $\boldsymbol{A}$ has full row rank; i.e., $m \leq n$, rank$(\boldsymbol{A}) = m$. (20) becomes

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \end{bmatrix} \Longleftrightarrow \tilde{\boldsymbol{x}}_1 = \tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{y}}, \quad \tilde{\boldsymbol{x}}_2 \text{ can be arbitrary.} \tag{23}$$

By transforming $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ back to $\boldsymbol{x}$ and $\boldsymbol{y}$, resp., the following results are concluded:

1) $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ has a solution.

2) Any

$$\boldsymbol{x} = \boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{U}^H\boldsymbol{y} + \boldsymbol{V}_2\boldsymbol{\alpha}, \quad \boldsymbol{\alpha} \in \mathbb{C}^{n-m}, \tag{24}$$

is a solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

Systems of Linear Equations (cont'd)

Case 3: Suppose that $\mathbf{A}$ is rank deficient; i.e., $r = \text{rank}(\boldsymbol{A}) < \min\{m, n\}$. By the same proof as above, the following results can be verified:

1) $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ has a solution if and only if $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{U}_1) = \mathcal{R}(\boldsymbol{A})$.

2) Any

$$\boldsymbol{x} = \boldsymbol{V}_1 \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{U}_1^H \boldsymbol{y} + \boldsymbol{V}_2 \boldsymbol{\alpha}, \quad \boldsymbol{\alpha} \in \mathbb{C}^{n-r}, \tag{25}$$

is a solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

Remark 2

The matrix $\boldsymbol{V}_1 \tilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{U}_1^H$ *is known as the* **pseudo-inverse** *of* $\boldsymbol{A}$.

Table of Contents

Low-Rank Matrix Approximation

Aim: Given a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ and a positive integer k, find a matrix $\boldsymbol{B} \in \mathbb{C}^{m\times n}$, with rank$(\boldsymbol{B}) < k$, such that $\boldsymbol{B}$ best approximates $\boldsymbol{A}$.

1) **Applications**: principal component analysis, latent semantic indexing (for discovering similarities between text documents), dimensionality reduction, data compression, $\cdots$

2) **Heuristic**: Let $\boldsymbol{A} = \sum_{i=1}^{p} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H$ be the SVD of $\boldsymbol{A}$ (see (4)), and denote

$$\boldsymbol{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H \tag{26}$$

as a truncated SVD of $\boldsymbol{A}$, where $j = 1, \cdots, p$. Choose $\boldsymbol{B} = \boldsymbol{A}_k$.

- Just engineering intuition, possibly no theory to begin with.

Table of Contents

Toy Application Example: Image Compression

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ whose $(i,j)^{\text{th}}$ element a_{ij} stores the $(i,j)^{\text{th}}$ pixel of an image.

(a) original image, size= 102 × 1347

Toy Application Example: Image Compression

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ whose $(i,j)^{\rm th}$ element a_{ij} stores the $(i,j)^{\rm th}$ pixel of an image.

(a) original image, size= 102 × 1347

Truncated SVD: store $\{\sigma_i, \boldsymbol{u}_i, \boldsymbol{v}_i\}_{i=1}^{k}$ instead of the full $\boldsymbol{A}$, and recover by $\boldsymbol{B} = \boldsymbol{A}_k$.

(b) truncated SVD, k= 5

(c) truncated SVD, k= 10

(d) truncated SVD, k= 20

Example: Principal Component Analysis (PCA)

Aim: Given a set of data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^m$, perform a low-dimensional representation

$$\boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{c}_i + \boldsymbol{\mu} + \boldsymbol{e}_i,\ i = 1, \cdots, n, \tag{27}$$

where

- $\boldsymbol{Q} \in \mathbb{R}^{m \times k}$ is a basis matrix; $\boldsymbol{c}_i \in \mathbb{R}^k$ is the corresponding coefficient for $\boldsymbol{x}_i$;
- $\boldsymbol{\mu} \in \mathbb{R}^m$ is the base (or mean in statistics terms);
- $\boldsymbol{e}_i \in \mathbb{R}^m$ is the representation error;
- $k < \min\{m, n\}$ is the dimension of the desired representation and is given.

Example: Principal Component Analysis (PCA)

Aim: Given a set of data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^m$, perform a low-dimensional representation

$$\boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{c}_i + \boldsymbol{\mu} + \boldsymbol{e}_i, \ i = 1, \cdots, n, \tag{27}$$

where

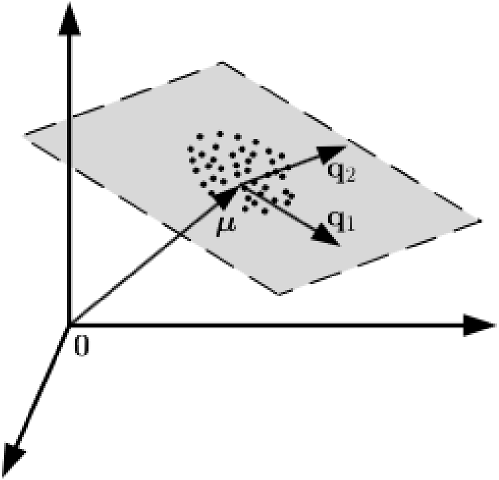
- $\boldsymbol{Q} \in \mathbb{R}^{m \times k}$ is a basis matrix; $\boldsymbol{c}_i \in \mathbb{R}^k$ is the corresponding coefficient for $\boldsymbol{x}_i$;
- $\boldsymbol{\mu} \in \mathbb{R}^m$ is the base (or mean in statistics terms);
- $\boldsymbol{e}_i \in \mathbb{R}^m$ is the representation error;
- $k < \min\{m, n\}$ is the dimension of the desired representation and is given.

1) The problem is to determine $\boldsymbol{Q}$, $\{\boldsymbol{c}_i\}$, and $\boldsymbol{\mu}$ from $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$.
2) Let $\boldsymbol{C} \triangleq [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n]$, $\boldsymbol{X} \triangleq [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$, $\boldsymbol{E} \triangleq [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n]$. We can write

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{C} + \boldsymbol{\mu}\boldsymbol{1}^T + \boldsymbol{E}. \tag{28}$$

Let $\boldsymbol{B} \triangleq \boldsymbol{Q}\boldsymbol{C}$. Since rank$(\boldsymbol{B}) \leq k$, the low-dimensional representation problem is closely related to the low-rank matrix approximation problem.

Example: PCA (cont'd)



Example: PCA (cont'd)

PCA solution: Let

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n. \tag{29}$$

Also, $\boldsymbol{Y} = [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \cdots, \boldsymbol{x}_N - \bar{\boldsymbol{x}}]$, and conduct SVD $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$.

Choose

$$\boldsymbol{\mu} = \bar{\boldsymbol{x}}, \tag{30}$$

which is indeed the sample mean of $\{\boldsymbol{x}_n\}$, and

$$\boldsymbol{Q} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k], \tag{31}$$

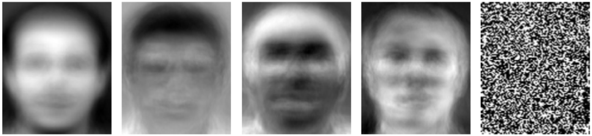
which is the first k left singular vectors of $\{\boldsymbol{x}_n - \bar{\boldsymbol{x}}\}$.

Toy Demo: Dimensionality Reduction of a Face Image Dataset

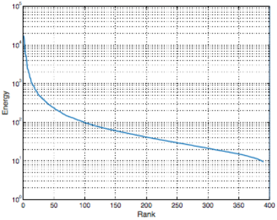
Figure 1: A face image dataset.

Image size $= 112 \times 92$, number of face images $= 400$. Each $\boldsymbol{x}_i$ is the vectorization of one face image, leading to $m = 112 \times 92 = 10304$ and $n = 400$.

Toy Demo: Dimensionality Reduction of a Face Image Dataset (cont'd)



Mean face | 1st principal left singular vector | 2nd principal left singular vector | 3rd principal left singular vector | 400th left singular vector



Energy Concentration

Figure 2: PCA of a face image dataset.

Table of Contents

Low-Rank Matrix Approximation

Question: is the truncated SVD theoretically sound, or is it just a heuristic?

Low-Rank Matrix Approximation

Question: is the truncated SVD theoretically sound, or is it just a heuristic?

Consider the low-rank matrix approximation formulation below.
Problem: Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a positive integer k, solve

$$\min_{\boldsymbol{B}=\mathbb{C}^{m\times n},\, \mathrm{rank}(\boldsymbol{B})\leq k} \|\boldsymbol{A}-\boldsymbol{B}\|_F. \tag{32}$$

Low-Rank Matrix Approximation

Question: is the truncated SVD theoretically sound, or is it just a heuristic?

Consider the low-rank matrix approximation formulation below.
Problem: Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a positive integer k, solve

$$\min_{\boldsymbol{B}=\mathbb{C}^{m\times n},\ \mathrm{rank}(\boldsymbol{B})\le k} \|\boldsymbol{A}-\boldsymbol{B}\|_F. \tag{32}$$

1) The solution to (32) is the best rank-k approximation to $\boldsymbol{A}$ in the sense of attaining the minimum root-mean-square error.
2) Does not seem to be easy to solve at first look, a non-convex optimization problem.
3) As it turns out, (32) can be solved by the truncated SVD.

Low-Rank Matrix Approximation (cont'd)

Theorem 2 (F-Norm Approximation)

Given any $\boldsymbol{A} = \mathbb{C}^{m\times n}$ *and* $k \in \{1, \cdots, p\}$, *the truncated SVD* $\boldsymbol{A}_k$ *is an optimal solution to Problem* (32).

Low-Rank Matrix Approximation (cont'd)

Theorem 2 (F-Norm Approximation)

Given any $\boldsymbol{A} = \mathbb{C}^{m\times n}$ and $k \in \{1, \cdots, p\}$, the truncated SVD $\boldsymbol{A}_k$ is an optimal solution to Problem (32).

The following result also holds.

Theorem 3 (2-Norm Approximation)

Given any $\boldsymbol{A} = \mathbb{C}^{m\times n}$ and $k \in \{1, \cdots, p\}$, the truncated SVD $\boldsymbol{A}_k$ is an optimal solution to

$$\min_{\boldsymbol{B}=\mathbb{C}^{m\times n},\ \mathit{rank}(\boldsymbol{B})\leq k} \|\boldsymbol{A}-\boldsymbol{B}\|_2 \tag{33}$$

- There is more than one way to prove Theorem 2. One way is to use singular value inequalities.

The Eckhart-Young Theorem[1]

Theorem 4 (The Eckhart-Young Theorem)

If $k < r = \mathrm{rank}(\boldsymbol{A})$ *and*

$$\boldsymbol{A}_k \triangleq \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H, \tag{34}$$

then,

$$\min_{\mathrm{rank}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2 = \|\boldsymbol{A} - \boldsymbol{A}_k\|_2 = \sigma_{k+1}. \tag{35}$$

[1] G. H. Golub and C. F. Van Load, *Matrix Computations*, 4th Ed., The John Hopkins University Press, 2013. (on page 79)

The Eckhart-Young Theorem[1]

Theorem 4 (The Eckhart-Young Theorem)

If $k < r = \mathrm{rank}(\boldsymbol{A})$ *and*

$$\boldsymbol{A}_k \triangleq \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H, \tag{34}$$

then,

$$\min_{\mathrm{rank}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2 = \|\boldsymbol{A} - \boldsymbol{A}_k\|_2 = \sigma_{k+1}. \tag{35}$$

Remark 3

- *Eq.* (35) *implies that the smallest singular value of* $\boldsymbol{A}$ *is the* 2*-norm distance of* $\boldsymbol{A}$ *to the set of all rank-deficient matrices.*
- *The matrix* $\boldsymbol{A}_k$ *defined in* (34) *is the closest rank-*k *matrix to* $\boldsymbol{A}$ *in the sense of* 2*-norm.*

[1] G. H. Golub and C. F. Van Load, *Matrix Computations*, 4th Ed., The John Hopkins University Press, 2013. (on page 79)

Weyl's Inequality

There is a rich collection of results concerning singular value inequalities, and here we show one.

Theorem 5 (Weyl's inequality)

Let $\boldsymbol{A} = \mathbb{C}^{m\times n}$ *and* $\boldsymbol{B} = \mathbb{C}^{m\times n}$ *be given, and let* $p = \min\{m, n\}$. *Then,*

$$\sigma_{i+j-1}(\boldsymbol{A} + \boldsymbol{B}) \leq \sigma_i(\boldsymbol{A}) + \sigma_j(\boldsymbol{B}), \tag{36}$$

for any $i, j \in \{1, \cdots, p\}$ *and* $i + j \leq p + 1$.

Weyl's Inequality

There is a rich collection of results concerning singular value inequalities, and here we show one.

Theorem 5 (Weyl's inequality)

Let $\boldsymbol{A} = \mathbb{C}^{m\times n}$ *and* $\boldsymbol{B} = \mathbb{C}^{m\times n}$ *be given, and let* $p = \min\{m, n\}$. *Then,*

$$\sigma_{i+j-1}(\boldsymbol{A}+\boldsymbol{B}) \leq \sigma_i(\boldsymbol{A}) + \sigma_j(\boldsymbol{B}), \tag{36}$$

for any $i, j \in \{1, \cdots, p\}$ *and* $i + j \leq p + 1$.

- Theorems 2-3 can be easily shown via Weyl's inequality.
- Weyl's inquality is useful in understanding perturbations of singular values. For example, as a special case of Weyl's inequality, we have

$$|\sigma_i(\boldsymbol{A}+\boldsymbol{E}) - \sigma_i(\boldsymbol{A})| \leq \sigma_1(\boldsymbol{E}), \text{ for } i = 1, \cdots, p \tag{37}$$

where $\boldsymbol{E}$ denotes a perturbation.

Table of Contents

System Model

- D_j: document (or web page) of interest, $j = 1, \cdots, n$
- T_i: terms (or key words) of interest, $i = 1, \cdots, m$
- freq_{ij}: number of times that term T_i occurs in document D_j
- $\boldsymbol{d}_j = \begin{bmatrix} \text{freq}_{1j} & \text{freq}_{2j} & \cdots & \text{freq}_{mj} \end{bmatrix}^T$: document vector
- $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_n \end{bmatrix}$: a term-by-document matrix (sparse matrix)
- **Query vector**: $\boldsymbol{q}^T = \begin{bmatrix} q_1 & q_2 & \cdots & q_m \end{bmatrix}$, where

$$q_i = \begin{cases} 1, & \text{if term } T_i \text{ appears in the query,} \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

The Principle of Page Ranks

To measure how well a query $\boldsymbol{q}$ matches a document D_j , we check how close $\boldsymbol{q}$ is to $\boldsymbol{d}_j$ by computing the magnitude of

$$\cos\theta_j = \frac{\boldsymbol{q}^T \boldsymbol{d}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{d}_j\|_2} = \frac{\boldsymbol{q}^T \boldsymbol{A}\boldsymbol{e}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{A}\boldsymbol{e}_j\|_2}. \tag{39}$$

If $\cos\theta_j \geq \tau$ for some threshold tolerance τ, then document D_j is considered relevant and is returned to the user.

Furthermore, if the columns of $\boldsymbol{A}$ along with $\boldsymbol{q}$ are initially normalized to have unit length, then

$$|\boldsymbol{q}^T \boldsymbol{A}| = \begin{bmatrix} |\cos\theta_1| & |\cos\theta_2| & \cdots & |\cos\theta_n| \end{bmatrix} \tag{40}$$

provides the information that allows the search engine to rank the relevance of each document relative to the query.

The Principle of Page Ranks

To measure how well a query $\boldsymbol{q}$ matches a document D_j , we check how close $\boldsymbol{q}$ is to $\boldsymbol{d}_j$ by computing the magnitude of

$$\cos\theta_j = \frac{\boldsymbol{q}^T \boldsymbol{d}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{d}_j\|_2} = \frac{\boldsymbol{q}^T \boldsymbol{A}\boldsymbol{e}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{A}\boldsymbol{e}_j\|_2}. \tag{39}$$

If $\cos\theta_j \geq \tau$ for some threshold tolerance τ, then document D_j is considered relevant and is returned to the user.

Furthermore, if the columns of $\boldsymbol{A}$ along with $\boldsymbol{q}$ are initially normalized to have unit length, then

$$|\boldsymbol{q}^T \boldsymbol{A}| = \begin{bmatrix} |\cos\theta_1| & |\cos\theta_2| & \cdots & |\cos\theta_n| \end{bmatrix} \tag{40}$$

provides the information that allows the search engine to rank the relevance of each document relative to the query.

Remark 4 (How to determine the threshold tolerance τ?)

Selecting τ is part art and part science thats based on experimentation and desired performance criteria.

Applying the Truncated SVD

However, due to things like variation and ambiguity in the use of vocabulary, presentation style, and even the indexing process, there is a lot of 'noise' in $\boldsymbol{A}$, so the results computed as per (40) are nowhere near being an exact measure of how well query $\boldsymbol{q}$ matches the various documents.

To filter out some of this noise, the truncated SVD of $\boldsymbol{A}$ is applied, i.e.,

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H, \tag{41}$$

$$\boldsymbol{A}_k \triangleq \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H, \quad k < r. \tag{42}$$

Then, (39) becomes

$$\cos\theta_j = \frac{\boldsymbol{q}^T \boldsymbol{A}_k \boldsymbol{e}_j}{\|\boldsymbol{q}^T\|_2 \, \|\boldsymbol{A}_k \boldsymbol{e}_j\|_2}. \tag{43}$$

More Efficient Computation

Let

$$\boldsymbol{S}_k \triangleq \boldsymbol{D}_k \boldsymbol{V}_k^T = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_k \end{bmatrix}, \tag{44}$$

$$\|\boldsymbol{A}_k \boldsymbol{e}_j\|_2 \triangleq \|\boldsymbol{U}_k \boldsymbol{D}_k \boldsymbol{V}_k^T \boldsymbol{e}_j\|_2 = \|\boldsymbol{U}_k \boldsymbol{s}_j\|_2 = \|\boldsymbol{s}_j\|_2. \tag{45}$$

Then, (43) can be computed as

$$\cos\theta_j = \frac{\boldsymbol{q}^T \boldsymbol{U}_k \boldsymbol{s}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{s}_j\|_2}. \tag{46}$$

Since the vectors in $\boldsymbol{U}_k$ and $\boldsymbol{S}_k$ only need to be computed once (and they can be determined without computing the entire SVD), so (46) requires very little computation to process each new query.

More Efficient Computation

Let

$$\boldsymbol{S}_k \triangleq \boldsymbol{D}_k \boldsymbol{V}_k^T = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_k \end{bmatrix}, \tag{44}$$

$$\|\boldsymbol{A}_k \boldsymbol{e}_j\|_2 \triangleq \|\boldsymbol{U}_k \boldsymbol{D}_k \boldsymbol{V}_k^T \boldsymbol{e}_j\|_2 = \|\boldsymbol{U}_k \boldsymbol{s}_j\|_2 = \|\boldsymbol{s}_j\|_2. \tag{45}$$

Then, (43) can be computed as

$$\cos\theta_j = \frac{\boldsymbol{q}^T \boldsymbol{U}_k \boldsymbol{s}_j}{\|\boldsymbol{q}^T\|_2 \ \|\boldsymbol{s}_j\|_2}. \tag{46}$$

Since the vectors in $\boldsymbol{U}_k$ and $\boldsymbol{S}_k$ only need to be computed once (and they can be determined without computing the entire SVD), so (46) requires very little computation to process each new query.

Remark 5 (How to determine the order k?)

In practice, k *in* (42) *is chosen significantly less than* r*, since*

1) *variations in the use of vocabulary and the ambiguity of many words produces significant noise in* $\boldsymbol{A}$*;*
2) *numerical accuracy to compute* (46) *is not an important issue (knowing a cosine to two or three significant digits is sufficient).*

Table of Contents

1. Procrustes[2] Transformations

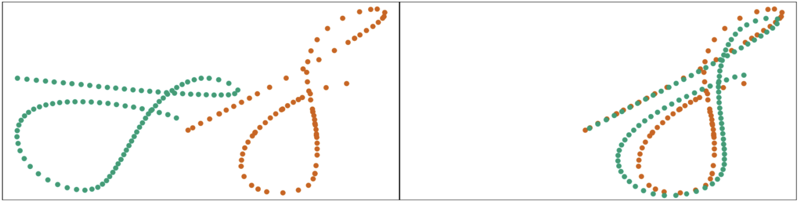


Figure 3: (Left panel) Two different digitized handwritten Ss, each represented by 96 corresponding points in $\mathbb{R}^2$. The green S has been deliberately rotated and translated for visual effect. (Right panel) A **Procrustes transformation** applies a translation and rotation to best match up the two set of point.

[2] Procrustes was an African bandit in Greek mythology, who stretched or squashed his visitors to fit his iron bed (eventually killing them).

Problem Formulation[3]

Definiton: A **Procrustes transformation** is a geometric transformation that involves only translation, rotation, uniform scaling, or a combination of these transformations. Hence, it may change the size, but not the shape of a geometric object. Mathematically, we aim to

$$\min_{\boldsymbol{\mu},\boldsymbol{R}} \left\| \boldsymbol{X}_2 - \left(\boldsymbol{X}_1 \mathbf{R} + \mathbf{1}\boldsymbol{\mu}^T \right) \right\|_F , \tag{47}$$

where

- $\boldsymbol{X}_1$, $\boldsymbol{X}_2$: $N \times p$ matrices of corresponding points,
- $\boldsymbol{R}$: orthonormal $p \times p$ matrix,
- $\boldsymbol{\mu}$: a p-dimension vector of location coordinates (e.g., $N = 96, p = 2$ in Fig. 3).

[3] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, 2nd Ed., Springer, 2017. (cf. Section 14.5.1)

Solution

Let $\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_2$ be the column mean vectors of the matrices, and $\tilde{\boldsymbol{X}}_1$ and $\tilde{\boldsymbol{X}}_2$ be the versions of these matrices with the means removed, respectively. Define the SVD:

$$\tilde{\boldsymbol{X}}_1^T \tilde{\boldsymbol{X}}_2 = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T. \tag{48}$$

Then, the solution to (47) is given by

$$\hat{\boldsymbol{R}} = \boldsymbol{U}\boldsymbol{V}^T, \tag{49}$$

$$\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}_2 - \hat{\boldsymbol{R}}\bar{\boldsymbol{x}}_1, \tag{50}$$

and the minimal distances is referred to as the **Procrustes distance**. From the form of the solution, we can center each matrix at its column centroid, and then ignore location completely. Hereafter we assume this is the case.

2. Procrustes Distance with Scaling

The **Procrustes distance with scaling** solves a slightly more general problem:

$$\min_{\beta, \boldsymbol{R}} \|\boldsymbol{X}_2 - \beta \boldsymbol{X}_1 \boldsymbol{R}\|_F \,, \tag{51}$$

where $\beta > 0$ is a positive scalar.

The solution for $\boldsymbol{R}$ in (51) is the same as (49), while the scaling factor is given by

$$\hat{\beta} = \frac{\mathrm{trace}(\boldsymbol{D})}{\|\boldsymbol{X}_1\|_F^2}, \tag{52}$$

where $\boldsymbol{D}$ is referred to as (48).

3. Procrustes Average

Related to Procrustes distance is the **Procrustes average** of a collection of L shapes, which solves the problem:

$$\min_{\{\boldsymbol{R}_\ell\}_1^L, \boldsymbol{M}} \sum_{\ell=1}^{L} \|\boldsymbol{X}_\ell \boldsymbol{R}_\ell - \boldsymbol{M}\|_F^2 \,, \tag{53}$$

that is, find the shape $\boldsymbol{M}$ closest in average squared Procrustes distance to all the shapes. The Procrustes average problem (53) can be solved by a simple alternating algorithm as shown in Algorithm 1.

Solution

Algorithm 1 Algorithm for solving the Procrustes average problem

1: Initialize $\boldsymbol{M} = \boldsymbol{X}_1$ (for example)
2: **repeat**
3: Solve the L Procrustes rotation problems with $\boldsymbol{M}$ fixed, yielding $\boldsymbol{X}'_\ell \leftarrow \boldsymbol{X}\hat{\boldsymbol{R}}_\ell$
4: Let $\boldsymbol{M} \leftarrow \frac{1}{L}\sum_{\ell=1}^{L} \boldsymbol{X}'_\ell$
5: **until** the criterion (53) converges.

An Example with Three Shapes

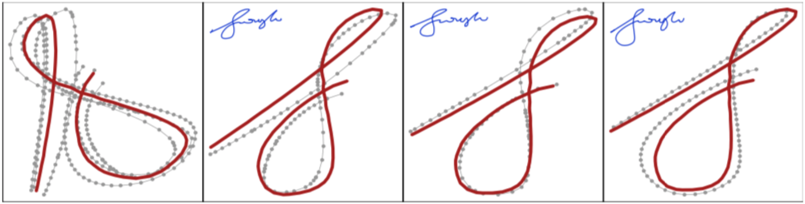


Figure 4: The Procrustes average of three versions of the leading S in Sureshs signatures. The left panel shows the preshape average, with each of the shapes $\boldsymbol{X}_\ell'$ in preshape space superimposed. The right three panels map the preshape $\boldsymbol{M}$ separately to match each of the original Ss.

4. Affine-Invariant Average

Most generally we can define the **affine-invariant average** of a set of shapes via

$$\min_{\{\boldsymbol{A}_\ell\}_1^L, \boldsymbol{M}} \sum_{\ell=1}^{L} \|\boldsymbol{X}_\ell \boldsymbol{A}_\ell - \boldsymbol{M}\|_F^2, \tag{54}$$

where $\boldsymbol{A}_\ell$ is any $p \times p$ nonsingular matrices. Here we require a standardization, such as $\boldsymbol{M}^T\boldsymbol{M} = \boldsymbol{I}$, to avoid a trivial solution.

The solution is attractive, and can be computed without iteration:

- Let $\boldsymbol{H}_\ell = \boldsymbol{X}_\ell \left(\boldsymbol{X}_\ell^T \boldsymbol{X}_\ell\right)^{-1} \boldsymbol{X}_\ell^T$ be the rank-p projection matrix defined by $\boldsymbol{X}_\ell$.
- $\boldsymbol{M}$ is the $N \times p$ matrix formed from the p eigenvectors of $\bar{\boldsymbol{H}} = \frac{1}{L}\sum_{\ell=1}^{L} \boldsymbol{H}_\ell$, pertaining to the p largest eigenvalues.

Remark 6 (Procrustes analysis in Matlab)

The Procrustes analysis can be implemented in Matlab by using the built-in function PROCRUSTES. *For more information, the interested reader is referred to* *https://en.wikipedia.org/wiki/Procrustes_analysis.*

Thank you
for your attention!

Kai Lu
E-mail: lukai86@mail.sysu.edu.cn
Web: http://seit.sysu.edu.cn/teacher/1801