# Singular Value Decomposition



## Singular Value Decomposition

*Eigendecomposition of symmetric matrices*

$\forall \boldsymbol{A} \in \mathbb{R}^{n \times n}$, there exist an orthonormal matrix $\boldsymbol{Q} \in \boldsymbol{R}^{n \times n}$ and a diagonal matrix $\boldsymbol{\Lambda} = \boldsymbol{diag}(\lambda_1, \cdots, \lambda_n)$

$$A = \boldsymbol{Q \Lambda Q}^T$$

*Singular Value Decomposition*

Extend the decomposition to **rectangular matrices**

$$\boldsymbol{X} = \boldsymbol{U S V^T}$$

# Applications in machine learning

- **Dimensionality Reduction**: SVD can be used for dimensionality reduction by reducing the rank of a matrix
- **Recommender Systems**: By factorizing the matrix using SVD, we can identify latent factors or features that capture underlying patterns and preferences.
- **Image Compression**: SVD is used in image compression techniques such as JPEG.
- **Latent Semantic Analysis**: By decomposing a term-document matrix using SVD, LSA can capture the latent semantic structure of the data
- **Principal Component Analysis (PCA)**: PCA is a SVD
- **Matrix Completion**: SVD-based techniques are used in matrix completion problems, where missing or incomplete data needs to be imputed.

# Existence of the SVD for general matrices

For any matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, there exist two orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{n \times n}$, $\boldsymbol{V} \in \mathbb{R}^{d \times d}$ and a nonnegative, "diagonal" matrix $\boldsymbol{S} \in \mathbb{R}^{n \times d}$ such that

$$\boldsymbol{X}_{n \times d} = \boldsymbol{U}_{n \times n} \boldsymbol{S}_{n \times d} \boldsymbol{V}_{d \times d}^T$$

where $\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$.

*In a vector form*

$$\boldsymbol{X}_{n \times d} = \sum_{j=1}^{r} S_{jj} \boldsymbol{u}_j \boldsymbol{v}_j^T$$



# Geometrical interpretation

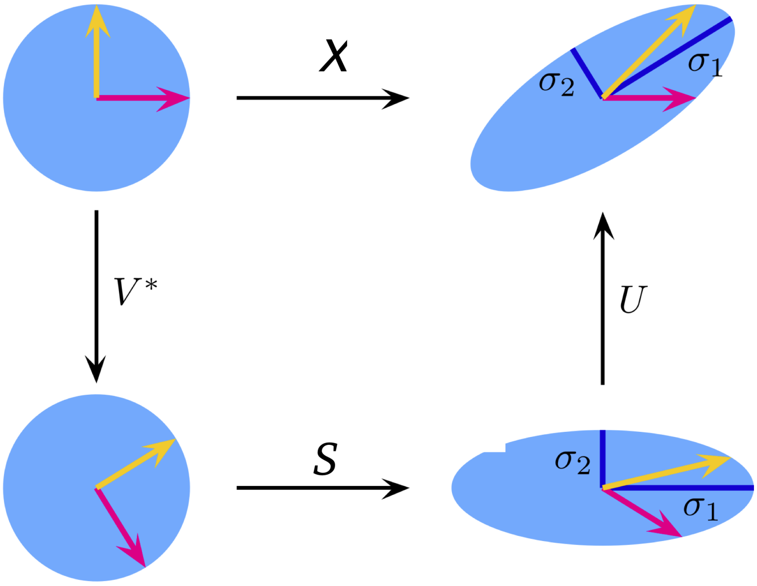
Given any matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ it defines a linear transformation:

$$f : \mathbb{R}^d \to \mathbb{R}^n, f(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{x}.$$

The linear transformation $f$ can be decomposed into three operations:

$$\underbrace{\boldsymbol{X}}_{linear\ transformation} \boldsymbol{x} = \underbrace{\boldsymbol{U}}_{rotation} \underbrace{\boldsymbol{S}}_{scaling} \underbrace{\boldsymbol{V}^T}_{rotation} \boldsymbol{x}$$

# Geometrical interpretation




$$X = U \cdot S \cdot V^*$$



# Different versions of SVD

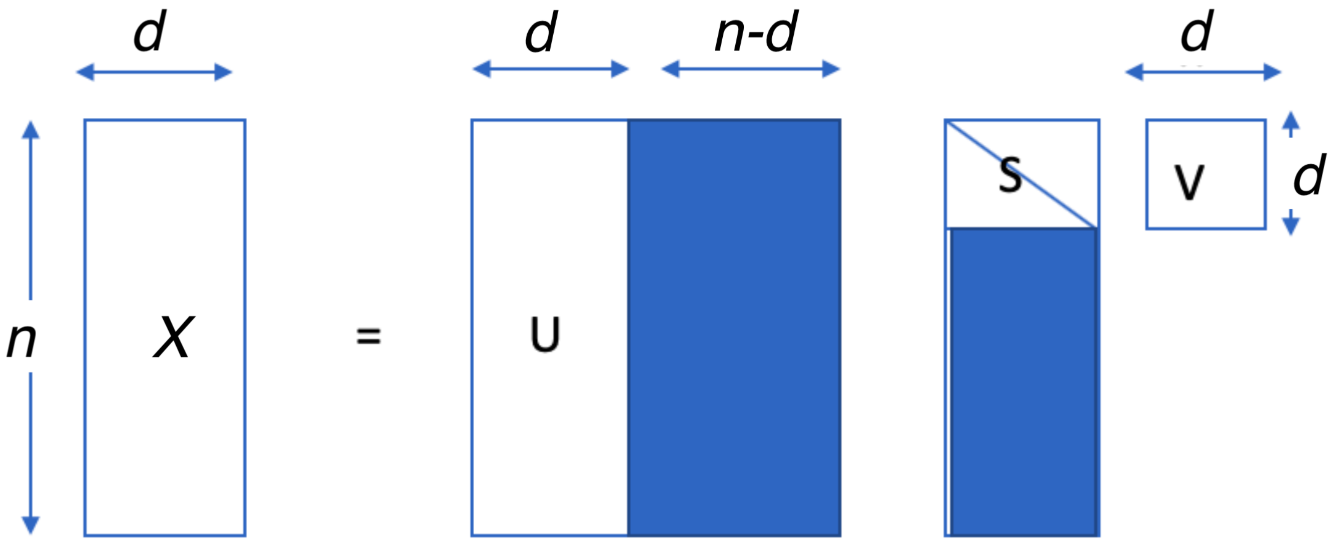
- Full SVD:

$$\boldsymbol{X}_{n\times d} = \boldsymbol{U}_{n\times n}\boldsymbol{S}_{n\times d}\boldsymbol{V}_{d\times d}^T$$

- Economy sized (thin, compact) SVD:

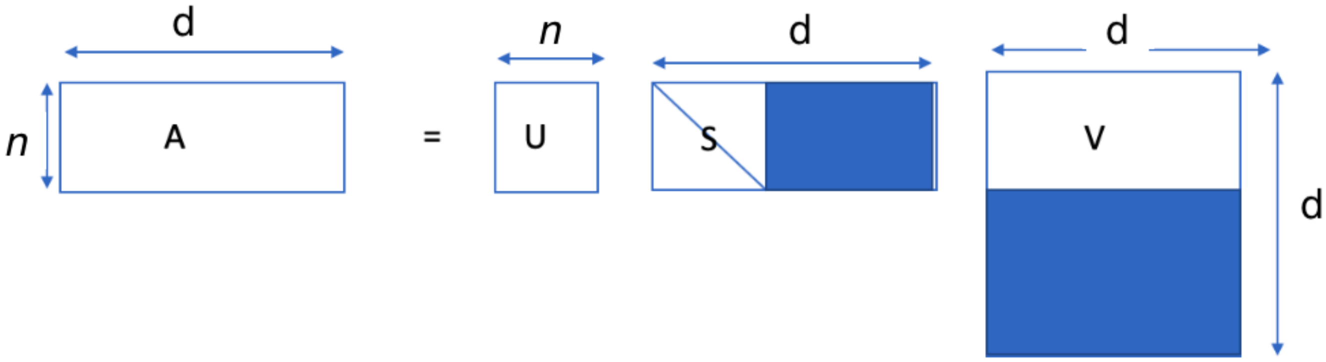
$$\boldsymbol{X}_{n\times d} = \boldsymbol{U}_{n\times r}\boldsymbol{S}_{r\times r}\boldsymbol{V}_{r\times d}^T$$

## SVD $n > d$






## SVD $n < d$



# Existence of the SVD

Consider $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ where $\boldsymbol{\Lambda} = diag(\lambda_1, \cdots, \lambda_d)$ with $\lambda_1 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_d$ (where $r = rank(\boldsymbol{X}) \leq d$).

Let $\sigma_i = \sqrt{\lambda_i}$ and correspondingly form the matrix

$$\boldsymbol{S}_{n\times d} = \begin{pmatrix} diag(\sigma_1, \cdots, \sigma_r) & 0_{r\times(d-r)} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(d-r)} \end{pmatrix}$$

Define also

$$\boldsymbol{u}_i = \frac{1}{\sigma_i}\boldsymbol{X}\boldsymbol{v}_i \in \mathbb{R}^n,$$

for each $1 \leq i \leq r$.

# Existence of the SVD

*Exercice*

It is easy to show that the $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r$ are orthonormal vectors.

*Completion if needed*

Choose $\boldsymbol{u}_{r+1}, \cdots, \boldsymbol{u}_n \in \mathbb{R}^n$ (through basis completion) such that

$$\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_n] \in \mathbb{R}^{n \times n}$$

is an orthogonal matrix.

It verifies

$$\boldsymbol{X}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{S},$$

i.e.,

# Existence of the SVD

$$\boldsymbol{X}[\boldsymbol{v}_1, \cdots \boldsymbol{v}_r \boldsymbol{v}_{r+1} \cdots \boldsymbol{v}_d] = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_r \boldsymbol{u}_{r+1} \boldsymbol{u}_n] \begin{pmatrix} diag(\sigma_1, \cdots, \sigma_r) & 0_r \\ 0_{(n-r)\times r} & 0_{(n-} \end{pmatrix}$$

Two possible cases:

- $\boldsymbol{1 \leq i \leq r : X v_i = \sigma_i u_i}$ by construction.
- $\boldsymbol{i > r : X v_i = 0}$, which is due to $\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{v}_i = \boldsymbol{C} \boldsymbol{v}_i = 0 \boldsymbol{v}_i = 0$.

Consequently, we have obtained that

$$\boldsymbol{X} = \boldsymbol{U} \boldsymbol{S} \boldsymbol{V}^T$$



# Properties

The linear application characterized by $\boldsymbol{X}$ has the following properties:

- $\boldsymbol{rank(X) = r}$ is the number of non zero singular values
- $\boldsymbol{kernel(X) = span(v_{r+1}, \cdots, v_n)}$
- $\boldsymbol{range(X) = span(u_1, \cdots, u_r)}$

# Low rank approximation of a matrix $\boldsymbol{X}$

## *Goal*

Approximate a given matrix $\boldsymbol{X}$ with a rank-k matrix, for a target rank k.

## *Motivations*

- Compression
- De-noising
- Matrix completion



# A first toy example

```
X<-matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),4,3,byrow=TRUE)
X.svd<-svd(X)
cat("Original matrix:\n")
```

```
Original matrix:
```

```
print(X)
```

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
```

```
k<-2
cat("Approximation of rank 2:\n")
```

```
Approximation of rank 2:
```

```
print(X.svd$u[,1:k]%*%diag(X.svd$d[1:k])%*%t(X.svd$v[,1:k]))
```

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
```

```
cat("A basis of the column space:\n")
```

```
A basis of the column space:
```

```r
print(X.svd$u[,1:k])
```

```
            [,1]        [,2]
[1,] -0.1408767 -0.82471435
[2,] -0.3439463 -0.42626394
[3,] -0.5470159 -0.02781353
[4,] -0.7500855  0.37063688
```

```r
cat("\nA basis of the kernel:\n")
```

```
A basis of the kernel:
```

```r
print(X.svd$u[,1:k])
```

```
            [,1]        [,2]
[1,] -0.1408767 -0.82471435
[2,] -0.3439463 -0.42626394
[3,] -0.5470159 -0.02781353
[4,] -0.7500855  0.37063688
```



# Illustration of svd in image compression

Example borrowed from rich-d-wilkinson.github.io

The 512 × 512 colour image is stored as three matrices R, B, G of the same dimension 512×512 giving the intensity of red, green, and blue for each pixel. Naively storing this matrix requires 5.7Mb.

```r
library(tiff)
library(rasterImage)
peppers<-readTIFF("../Silo-Images/Peppers.tiff")
plot(as.raster(peppers))
```

# Illustration of svd in image compression

Below the SVD of the three colour intensity matrices, and the view the image that results from using reduced rank versions with rank $k \in \{5, 30, 100, 300\}$

```r
svd_image <- function(im,k){
  s <- svd(im)
  Sigma_k <- diag(s$d[1:k])
  U_k <- s$u[,1:k]
  V_k <- s$v[,1:k]
  im_k <- U_k %*% Sigma_k %*% t(V_k)
   ## the reduced rank SVD produces some intensities <0 and >1.
  # Let's truncate these
  im_k[im_k>1]=1
  im_k[im_k<0]=0
  return(im_k)
}

par(mfrow=c(2,2), mar=c(1,1,1,1))


pepprssvd<- peppers
```

# Low rank approximation of a matrix $\boldsymbol{X}$

## *Frobenius norm*

The Frobenius norm of a matrix $\boldsymbol{X}$ is defined as

$$\|\boldsymbol{X}\|_F^2 = \sum_{ij} X_{ij}^2 = trace(\boldsymbol{X}^T\boldsymbol{X}) = \sum_{j=1}^{r} \sigma_j^2$$

## *Rank k matrix $\hat{\boldsymbol{X}}_k$*

Let

$$\hat{\boldsymbol{X}}_k = \sum_{j=1}^{k} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T$$

# Low rank approximation of a matrix

For any matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ with non null singular values $\boldsymbol{\sigma}_1 \geq \boldsymbol{\sigma}_2 \geq \cdots \geq \boldsymbol{\sigma}_r$

$$\hat{\boldsymbol{X}}_k = \arg \min_{\hat{X}:rank(\hat{X})=k} \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2$$

$$\min_{\hat{X}:rank(\hat{X})=k} \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 = \sum_{j=k+1}^{r} \sigma_j^2$$



# Proof

We have

$$\|X - X_k\|_F^2 = \left\| \sum_{i=k+1}^{n} \sigma_i u_i v_i^\top \right\|_F^2 = \sum_{i=k+1}^{n} \sigma_i^2$$

We need to show that if $Y_k = AB^\top$ where $A$ and $B$ have k columns then

$$\|X - X_k\|_F^2 = \sum_{i=k+1}^{n} \sigma_i^2 \leq \|X - Y_k\|_F^2.$$

# Proof

By the triangle inequality with the spectral norm, if $X = X' + X''$ then $\sigma_1(X) \leq \sigma_1(X') + \sigma_1(X'')$ .

Suppose $X'_k$ and $X''_k$ respectively denote the rank k approximation to $X'$ and $X''$ by SVD.

Then, for any $i, j \geq 1$

$$\begin{aligned}
\sigma_i(X') + \sigma_j(X'') &= \sigma_1(X' - X'_{i-1}) + \sigma_1(X'' - X''_{j-1}) \\
&\geq \sigma_1(X - X'_{i-1} - X''_{j-1}) \\
&\geq \sigma_1(X - X_{i+j-2}) \qquad \text{(since rank}(X'_{i-1} \\
&= \sigma_{i+j-1}(X).
\end{aligned}$$

# Proof

Since $\sigma_{k+1}(Y_k) = 0$, when $X' = X - Y_k$ and $X'' = Y_k$ we conclude that for $i \geq 1, j = k+1$

$\sigma_i(X - Y_k) + \underbrace{\sigma_{k+1}(Y_k)}_{0} \geq \sigma_{k+i}(X)$. Therefore,

$$\|X - Y_k\|_F^2 = \sum_{i=1}^{n} \sigma_i(X - Y_k)^2 \geq \sum_{i=k+1}^{n} \sigma_i(X)^2 = \|X - X_k$$



# Low rank approximation of a matrix and projection

If $rank(\hat{X}) = k$, then we can assume columns $\hat{X}_i$ of $\hat{\boldsymbol{X}} \in E_k = span\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k\}$ where $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k\}$ is a set of orthonormal vectors for the linear space of columns of $X_k$. First, observe that

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 = \sum_i \|X_i - \hat{X}_i\|^2$$

*Optimum solution is the orthogonal projection*

For each term $\|X_i - v\|_2^2$,the optimum solution is the projection of $X_i$ onto $E_k = span\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k\}$:

$$\hat{X}_i = \sum_{j=1}^{k} \langle X_i, \boldsymbol{w}_j \rangle \boldsymbol{w}_j = \Pi_{E_k} X_i.$$

where $\Pi_{E_k} = \sum_{j=1}^{k} \boldsymbol{w}_j \boldsymbol{w}_j^T$



# Projection on the orthogonal subspace

Consider $\Pi_{E_k^\perp}$ the projection matrix on the space orthogonal to $E_k$. More precisely, let us add $\boldsymbol{w}_{k+1}, \cdots, \boldsymbol{w}_n$ such that $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n$ form an orthonormal basis of $\mathbb{R}^n$. Then,

$$\Pi_{E_k^\perp} = \sum_{j=k+1}^{n} \boldsymbol{w}_j \boldsymbol{w}_j^T$$

$$\|X - \hat{X}\|_F^2 = \|X - \Pi_{E_k} X\|_F^2 = \|(I - \Pi_{E_k}) X\|_F^2 = \|\Pi_{E_k^\perp} X$$

# Relation to principal component analysis

*Warning*

$\boldsymbol{X}$ **is considered as centered**. This tranformation (cloud translation allows considerable simplification)

*Decomposition of* $\boldsymbol{X}$

Considering the orthonal projection on $\boldsymbol{E}_k$

$$\boldsymbol{X} = \Pi_{E_k}\boldsymbol{X} + \Pi_{E_k^\perp}\boldsymbol{X}$$



# Criterion

$$\|\boldsymbol{X}\|_F^2 = \underbrace{\|\Pi_{E_k}\boldsymbol{X}\|_F^2}_{approximation} + \underbrace{\|\Pi_{E_k^\perp}\boldsymbol{X}\|_F^2}_{error}$$

In terms of intertia, PCA maximizes the projected inertia (approximation) while minimizing the ditances to the space of projection (error):

$$\boldsymbol{I}_T = \boldsymbol{I}_E + \boldsymbol{I}_{E_k^\perp}$$

# Best low rank approximation

$$\hat{\boldsymbol{X}}_k = \Pi_{E_k}\boldsymbol{X} = \sum_{j=1}^{k} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T = \boldsymbol{U}_{\bullet,1:k}\boldsymbol{S}_{1:k,1:k}\boldsymbol{V}_{\bullet,1:k}^T$$

where $\boldsymbol{X} = \underbrace{\boldsymbol{U}}_{n\times n}\underbrace{\boldsymbol{S}}_{n\times k}\underbrace{\boldsymbol{V}}_{d\times d}^T$



# Different views of the approximation

The approximation

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2$$

can be considered in multiple ways:

- approximation of the row
- approximation of the columns

*Notations*

If $\boldsymbol{X}$ is a data table,

- each row $\boldsymbol{x}_i^T$ is a description of an individual
- each colum $\boldsymbol{X}_j$ is variable describing $\boldsymbol{n}$ individuals



# Rows approximation (projection of the individuals)

Transposing the matrix the best low rank approximation becomes

$$\hat{\boldsymbol{X}^T}_k = \Pi_{F_k}\boldsymbol{X}^T = \sum_{j=1}^{k} \sigma_j \boldsymbol{v}_j \boldsymbol{u}_j^T = \boldsymbol{V}_{\bullet,1:k}\boldsymbol{S}_{1:k,1:k}\boldsymbol{U}_{\bullet,1:k}^T$$

where $\boldsymbol{F}_k = \boldsymbol{span}\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\}$

# The approximation error

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 = \|\boldsymbol{X}^T - \hat{\boldsymbol{X}}^T\|_F^2 = \sum_i \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2^2$$

Each row $\boldsymbol{x}_i$ is approximated by

$$\hat{\boldsymbol{x}}_i = \Pi_{F_k}\boldsymbol{x}_i = V_{F_k}V_{F_k}^T\boldsymbol{x}_i$$

where $\boldsymbol{V}_{F_k}$ is the matrix composed of the vectors defining $\boldsymbol{F}_k$.



# Projection of the variables

$\boldsymbol{\Pi}_{E_k}$ is the projection matrix on $\boldsymbol{E} = \boldsymbol{span}(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k)$

$$\Pi_E = \boldsymbol{U}_{E_k}\boldsymbol{U}_{E_k}^T$$

and

$$\boldsymbol{\Pi}_{E_k}\boldsymbol{X} = \boldsymbol{U}_{1:n,1:k}\underbrace{\boldsymbol{U}_{1:n,1:k}^T\boldsymbol{U}_{1:n,1:n}}_{(I_k, 0_{k,n-k})}\boldsymbol{S}_{1:n,1:k}\boldsymbol{V}_{1:d,1:d}^T = \boldsymbol{U}_{1:n,1:k}\boldsymbol{S}_{1:k}$$

# k first principal components

$$\boldsymbol{C}_{1:n,1:k} = \boldsymbol{U}_{E_k}\boldsymbol{S}_{1:k,1:k}$$

where $\boldsymbol{S}_{1:k,1:k} = \boldsymbol{diag}(\sigma_1, \cdots, \sigma_k)$.

The principal component are the coordinates of the projection of the rows of $\boldsymbol{X}$ on $\boldsymbol{F}_k$:

$$\boldsymbol{C}_{1:n,1:k} = \boldsymbol{X}\boldsymbol{V}_{1:d,1:k}$$



# Percentage of information

We have $\boldsymbol{C}^T_{\bullet,1:k}\boldsymbol{C}_{\bullet,1:k} = S^2_{1:k,1:k} = \boldsymbol{diag}(\sigma_1^2, \cdots, \sigma_k^2)$, thus

$$\|\boldsymbol{C}_{\bullet,1:k}\|_F^2 = \sum_{j=1}^{k} \sigma_j^2$$

and

$$\frac{\|\boldsymbol{C}_{\bullet,1:k}\|_F^2}{\|\boldsymbol{X}\|_F^2} = \frac{\sum_{j=1}^{k} \sigma_j^2}{\sum_{j=1}^{d} \sigma_j^2} \in [0, 1]$$

# Correlations

$$\widehat{cor}(\boldsymbol{X}_{\bullet,j}, \boldsymbol{C}_{\bullet,k}) = \frac{X_{\bullet,j}^T \boldsymbol{C}_{\bullet,k}}{\|X_{\bullet,j}\| \|\boldsymbol{C}_{\bullet,k}\|} = \cos\left(\widehat{\boldsymbol{X}_{\bullet,j}, \boldsymbol{C}_{\bullet,k}}\right)$$



# Duality

It is easy to show that

- the columns of $\boldsymbol{V}$ are the eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$
- the columns of $\boldsymbol{U}$ are the eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$

Thus the principal component of $\boldsymbol{X}^T\boldsymbol{X}$ are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and vice-versa