

Singular Value Decomposition (SVD)

Predictive Modeling & Statistical Learning

Gaston Sanchez

CC BY-SA 4.0

Matrix Decompositions

Decompositions

Matrix decompositions, also known as matrix factorizations

$$\mathbf{M} = \mathbf{AB} \quad \text{or} \quad \mathbf{M} = \mathbf{ABC}$$

are a means of expressing a matrix as a product of usually **two** or **three** **simpler** matrices.

Importance of Decompositions

What for?

Matrix decompositions make it easier to study the properties of matrices. Likewise, many computation tasks become easier with decompositions.

They play a relevant role in multivariate data analysis. Often, the solution to many techniques are obtained (or derived) from a matrix decomposition.

Decompositions: What for?

- ▶ solving systems of linear equations
- ▶ inverting a matrix
- ▶ analyzing numerical stability of a system
- ▶ understanding the structure of data
- ▶ finding **basis** for column space (or row space) of a matrix

Some Assumptions

Real Matrices

We will assume all matrices to be **real** matrices, i.e. matrices containing elements in the set of Real numbers.

Dimensions $n \geq p$

Unless otherwise stated, we will also assume matrices with more rows than columns.

Decompositions

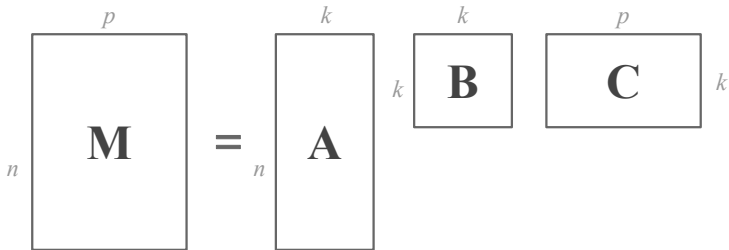
A matrix decomposition can be described by an equation:

$$\mathbf{M} = \mathbf{A}\mathbf{B}\mathbf{C}$$

where the dimensions of the matrices are as follows:

- ▶ \mathbf{M} is $n \times p$ (assume $n > p$)
- ▶ \mathbf{A} is $n \times k$ (usually $k < p$)
- ▶ \mathbf{B} is $k \times k$ (usually diagonal)
- ▶ \mathbf{C} is $k \times p$

Matrix Decomposition



Interpreting Decompositions

The equation that describes a decomposition:

$$\mathbf{M} = \mathbf{ABC}$$

- ▶ does not explain how to compute one
- ▶ does not explain how such decomposition can reveal the structures implicit in a data matrix.
- ▶ Seeing how a matrix decomposition reveals structure in a dataset is more complicated
- ▶ Each decomposition reveals a different kind of implicit structure

Types of matrices

Two types of matrices

We concentrate on the **two types of matrices** important in statistics:

- ▶ general **rectangular** matrices used to represent data tables.
- ▶ **positive semi-definite** matrices used to represent covariance matrices, correlation matrices, and any matrix that results from a crossproduct.

Two Special Decompositions

SVD and EVD

There are many types of matrix decompositions but for now we are going to consider only two:

- ▶ Singular Value Decomposition (SVD)
- ▶ Eigen-Value Decomposition (EVD)

SVD

Singular Value Decomposition

- ▶ One of the most important decompositions in matrix algebra
- ▶ Can be applied to **any** rectangular matrix
- ▶ ANY: rectangular or square, singular or nonsingular.

Singular Value Decomposition

An $n \times p$ matrix \mathbf{M} can be decomposed as:

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

where

- ▶ \mathbf{U} is a $n \times p$ column *orthonormal* matrix containing the **left singular vectors**
- ▶ \mathbf{D} is a $p \times p$ **diagonal** matrix containing the **singular values** of \mathbf{M}
- ▶ \mathbf{V} is a $p \times p$ column **orthonormal** matrix containing the **right singular vectors**

SVD

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^{\top}$$

$$\mathbf{M} = \begin{bmatrix} u_{11} & \cdots & u_{1p} \\ u_{21} & \cdots & u_{2p} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{np} \end{bmatrix} \begin{bmatrix} l_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_p \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{p1} \\ \vdots & \ddots & \vdots \\ v_{1p} & \cdots & v_{pp} \end{bmatrix}$$

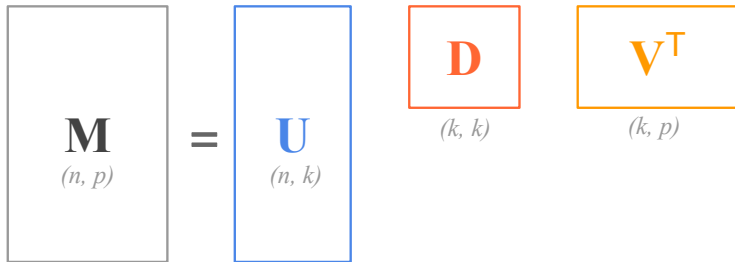
SVD Diagram

The diagram illustrates the Singular Value Decomposition (SVD) of a matrix M . It consists of four rectangular boxes arranged horizontally, separated by an equals sign. The first box, with a black border, contains the matrix M in black font with dimensions (n, p) below it. The second box, with a blue border, contains the matrix U in blue font with dimensions (n, p) below it. The third box, with a red border, contains the matrix D in red font with dimensions (p, p) below it. The fourth box, with an orange border, contains the matrix V^T in orange font with dimensions (p, p) below it.

$$\begin{matrix} \mathbf{M} \\ (n, p) \end{matrix} = \begin{matrix} \mathbf{U} \\ (n, p) \end{matrix} \begin{matrix} \mathbf{D} \\ (p, p) \end{matrix} \begin{matrix} \mathbf{V}^T \\ (p, p) \end{matrix}$$

When M is of full rank p

SVD Diagram



When M is of rank $k < p$

SVD

Singular Value Decomposition

We can think of the SVD structure as *the basic structure of a matrix*. What do we mean by “basic”? Well, this has to do with what each of the matrices $\mathbf{U}\mathbf{D}\mathbf{V}^T$ represent.

- ▶ \mathbf{U} is the **orthonormalized** matrix which is the most basic component. It's like the skeleton of the matrix.
- ▶ \mathbf{D} is referred to as the *spectrum* and it is a **scale component**.
- ▶ \mathbf{V} is an **orientation** component, also referred to as the *rotation* matrix.

SVD

- ▶ \mathbf{U} is unitary, and its columns form a basis for the space spanned by the columns of \mathbf{M} .

$$\mathbf{U}^T \mathbf{U} = \mathbf{I}_p$$

- ▶ \mathbf{V} is unitary, and its columns form a basis for the space spanned by the rows of \mathbf{M} .

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}_p$$

- ▶ \mathbf{D} has non-negative real numbers on the diagonal (assuming \mathbf{M} is real).

SVD in R

svd() in R

svd() function

R provides the function `svd()` to perform a singular value decomposition of a given matrix

svd() output

A list with the following components

- `d` a vector containing the singular values
- `u` a matrix whose columns contain the left singular vectors
- `v` a matrix whose columns contain the right singular vectors

SVD example in R

```
# X matrix
set.seed(22)
X = matrix(rnorm(20), 5, 4)

# singular value decomposition
SVD = svd(X)

# elements returned by svd()
names(SVD)

## [1] "d" "u" "v"

# vector of singular values
(d = SVD$d)

## [1] 3.9516353 2.0223602 1.4748193 0.4324292
```

SVD example in R (con't)

```
# matrix of left singular vectors
```

```
(U = SVD$u)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] -0.4251177 -0.53913435 -0.7232572  0.00979433
## [2,]  0.5268694 -0.76862769  0.2860048  0.05610045
## [3,]  0.5752546  0.04999546 -0.4421464  0.13107213
## [4,]  0.2215220  0.05272644 -0.1702161 -0.95123359
## [5,] -0.4021114 -0.33655016  0.4130778 -0.27337073
```

```
# matrix of right singular vectors
```

```
(V = SVD$v)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,]  0.5708354 -0.7406782  0.33862988  0.1042716
## [2,] -0.2741800 -0.5295008 -0.76797328  0.2338189
## [3,]  0.2772481  0.3206239 -0.04462207  0.9046229
## [4,]  0.7225689  0.2611992 -0.54180782 -0.3407543
```

SVD example in R (con't)

```
# U orthonormal (U'U = I)
```

```
t(U) %*% U
```

```
##           [,1]           [,2]           [,3]           [,4]
## [1,] 1.000000e+00  1.387779e-16  2.775558e-17  0.000000e+00
## [2,] 1.387779e-16  1.000000e+00 -2.775558e-17 -8.326673e-17
## [3,] 2.775558e-17 -2.775558e-17  1.000000e+00  5.551115e-17
## [4,] 0.000000e+00 -8.326673e-17  5.551115e-17  1.000000e+00
```

```
# V orthonormal (V'V = I)
```

```
t(V) %*% V
```

```
##           [,1]           [,2]           [,3]           [,4]
## [1,] 1.000000e+00 -1.110223e-16 -5.551115e-17  1.110223e-16
## [2,] -1.110223e-16  1.000000e+00  8.326673e-17  1.942890e-16
## [3,] -5.551115e-17  8.326673e-17  1.000000e+00 -8.326673e-17
## [4,] 1.110223e-16  1.942890e-16 -8.326673e-17  1.000000e+00
```

SVD example in R (con't)

```
# X equals U D V'  
U %*% diag(d) %*% t(v)
```

```
##           [,1]      [,2]      [,3]      [,4]  
## [1,] -0.5121391  1.85809239 -0.76390728 -0.9221536  
## [2,]  2.4851837 -0.06602641  0.08196190  0.8615624  
## [3,]  1.0078262 -0.16276495  0.74302828  2.0029422  
## [4,]  0.2928146 -0.19986068 -0.08402219  0.9365510  
## [5,] -0.2089594  0.30056173 -0.79289452 -1.6157349
```

```
# compare to X  
X
```

```
##           [,1]      [,2]      [,3]      [,4]  
## [1,] -0.5121391  1.85809239 -0.76390728 -0.9221536  
## [2,]  2.4851837 -0.06602641  0.08196190  0.8615624  
## [3,]  1.0078262 -0.16276495  0.74302828  2.0029422  
## [4,]  0.2928146 -0.19986068 -0.08402219  0.9365510  
## [5,] -0.2089594  0.30056173 -0.79289452 -1.6157349
```


SVD and Cross-products

Data Matrix

Data

The analyzed data can be expressed in matrix format \mathbf{X} :

$$\mathbf{X}_{n \times p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

- ▶ n objects in the rows
- ▶ p variables in the columns

Relation of SVD and Cross-Product Matrices

The cross-product matrix of columns of \mathbf{X} can be expressed as:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{D}^2 \mathbf{V}^T$$

Relation of SVD and Cross-Product Matrices

The cross-product matrix of columns can be expressed as:

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= (\mathbf{U}\mathbf{D}\mathbf{V}^T)^T(\mathbf{U}\mathbf{D}\mathbf{V}^T) \\ &= (\mathbf{V}\mathbf{D}\mathbf{U}^T)(\mathbf{U}\mathbf{D}\mathbf{V}^T) \\ &= \mathbf{V}\mathbf{D}(\mathbf{U}^T\mathbf{U})\mathbf{D}\mathbf{V}^T \\ &= \mathbf{V}\mathbf{D}^2\mathbf{V}^T\end{aligned}$$

Relation of SVD and Cross-Product Matrices

The cross-product matrix of rows of \mathbf{X} can be expressed as:

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{D}^2\mathbf{U}^T$$

Relation of SVD and Cross-Product Matrices

The cross-product matrix of rows can be expressed as:

$$\begin{aligned}\mathbf{XX}^T &= (\mathbf{UDV}^T)(\mathbf{UDV}^T)^T \\ &= (\mathbf{UDV}^T)(\mathbf{VDU}^T) \\ &= \mathbf{UD}(\mathbf{V}^T\mathbf{V})\mathbf{DU}^T \\ &= \mathbf{UD}^2\mathbf{U}^T\end{aligned}$$

Relation of SVD and Cross-Product Matrices

One of the interesting things about SVD is that \mathbf{U} and \mathbf{V} are matrices whose **columns** are **eigenvectors** of product moment matrices that are *derived* from \mathbf{X} . Specifically,

- ▶ \mathbf{U} is the matrix of eigenvectors of (symmetric) $\mathbf{X}\mathbf{X}^T$ of order $n \times n$
- ▶ \mathbf{V} is the matrix of eigenvectors of (symmetric) $\mathbf{X}^T\mathbf{X}$ of order $p \times p$

Of additional interest is the fact that **\mathbf{D} is a diagonal** matrix whose main diagonal entries are the **square roots of Λ** , the *common* matrix of eigenvalues of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$.



Rank Reduction

SVD Rank-Reduction Theorem

In terms of the diagonal elements l_1, l_2, \dots, l_r of \mathbf{D} , the columns $\mathbf{u}_1, \dots, \mathbf{u}_r$ of \mathbf{U} , and the columns $\mathbf{v}_1, \dots, \mathbf{v}_r$ of \mathbf{V} , the basic structure of \mathbf{X} may be written as

$$\mathbf{X} = l_1 \mathbf{u}_1 \mathbf{v}_1^T + l_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots + l_p \mathbf{u}_p \mathbf{v}_p^T$$

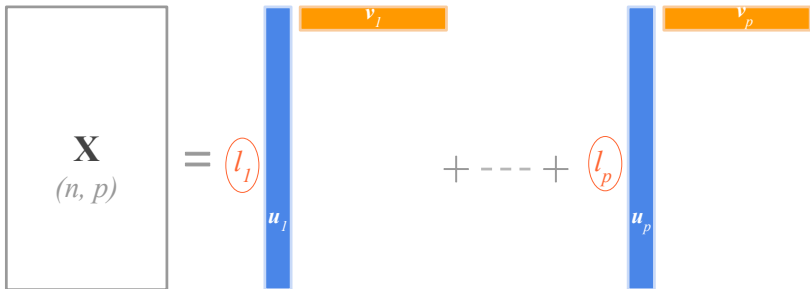
which shows that the matrix \mathbf{X} of rank p is a linear combination of r matrices of rank 1.

SVD Rank-Reduction Theorem

A very interesting and alternative way to represent the SVD is with the following formula:

$$\mathbf{X} = \sum_{k=1}^p l_k \mathbf{u}_k \mathbf{v}_k^T$$

SVD Diagram



SVD as sum of rank one matrices

SVD Rank-Reduction Theorem

SVD alternative formula:

$$\mathbf{X} = \sum_{k=1}^p l_k \mathbf{u}_k \mathbf{v}_k^T$$

- ▶ This expresses the SVD as a sum of p rank 1 matrices.
- ▶ This result is formalized in what is known as the **SVD theorem** described by Carl Eckart and Gale Young in 1936, and it is often referred to as the Eckart-Young theorem.
- ▶ This theorem applies to practically any arbitrary rectangular matrix.

SVD Rank-Reduction Theorem

What if you take $r < p$ terms?

$$\hat{\mathbf{X}} = \sum_{k=1}^r l_k \mathbf{u}_k \mathbf{v}_k^T$$

How would $\hat{\mathbf{X}}$ compare to \mathbf{X} ?

SVD Rank-Reduction Theorem

The SVD theorem of Eckart and Young is related to the important problem of approximating a matrix.

The basic result says that if \mathbf{X} is an $n \times p$ rectangular matrix, then the best r -dimensional approximation $\hat{\mathbf{X}}$ to \mathbf{X} is obtained by minimizing:

$$\min \quad \|\mathbf{X} - \hat{\mathbf{X}}\|^2$$

SVD Rank-Reduction Theorem

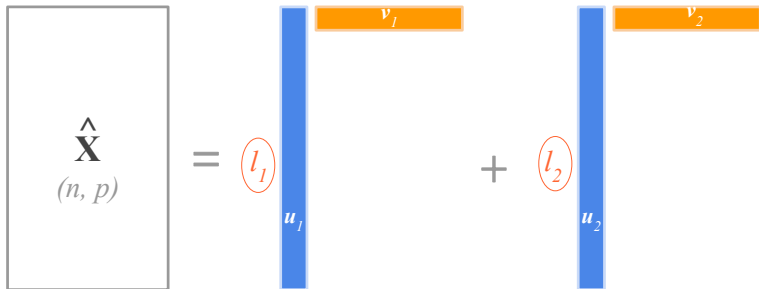
The minimization problem:

$$\min \quad \|\mathbf{X} - \hat{\mathbf{X}}\|^2$$

is a special type of approximation: a least squares approximation.

The solution is obtained by taking the first r elements of matrices \mathbf{U} , \mathbf{D} , \mathbf{V} so that $\hat{\mathbf{X}} = \mathbf{U}_r \mathbf{D}_r \mathbf{V}_r^T$

SVD rank-two approximation



The diagram illustrates the SVD rank-two approximation of a matrix $\hat{\mathbf{X}}$. On the left, a large rectangle contains the matrix $\hat{\mathbf{X}}$ with dimensions (n, p) below it. This is followed by an equals sign. To the right of the equals sign are two terms added together. The first term consists of a red circle containing l_1 next to a blue vertical rectangle labeled u_1 at its base, which is then multiplied by an orange horizontal rectangle labeled v_1 at its top. The second term is similar, with a red circle containing l_2 next to a blue vertical rectangle labeled u_2 at its base, multiplied by an orange horizontal rectangle labeled v_2 at its top.

$$\hat{\mathbf{X}}_{(n, p)} = l_1 u_1 v_1 + l_2 u_2 v_2$$

SVD as sum of two rank one matrices

SVD Rank-Reduction Theorem

The best 2-rank approximation of \mathbf{X} is given by:

$$\hat{\mathbf{X}} = l_1 \mathbf{u}_1 \mathbf{v}_1^T + l_2 \mathbf{u}_2 \mathbf{v}_2^T$$

The “information” contained in $n \times p$ values is compressed into $n \times 2$ values.