

## SINGULAR VALUE DECOMPOSITION

The Singular Value Decomposition (SVD) is a numerical technique that enables us to extend the idea of matrix diagonalization

$$A = X \Lambda X^{-1}$$

to any complex or real matrix.

The SVD decomposes a matrix  $A$  in a product of three matrices

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

where  $U$  and  $V$  are orthogonal.

The most significant differences between a proper diagonalization and the SVD are the following:

- First of all, usually  $U \neq V$  then,  $U$  and  $V$  are orthogonal while in general  $X$  is not. The SVD has many useful applications in data mining, signal processing and statistics because of its endless advantages.

For example, it allows to order the information contained in the matrix so that, loosely speaking the "dominating part" becomes visible.

This is due to the properties of orthogonal matrices. As a matter of fact :

$$\|A\|_F^2 = \|\Sigma\|_F^2 = \|\text{diag}(\sigma_1, \dots, \sigma_n)\|_F^2 = \sigma_1^2 + \dots + \sigma_n^2$$

## THE DECOMPOSITION

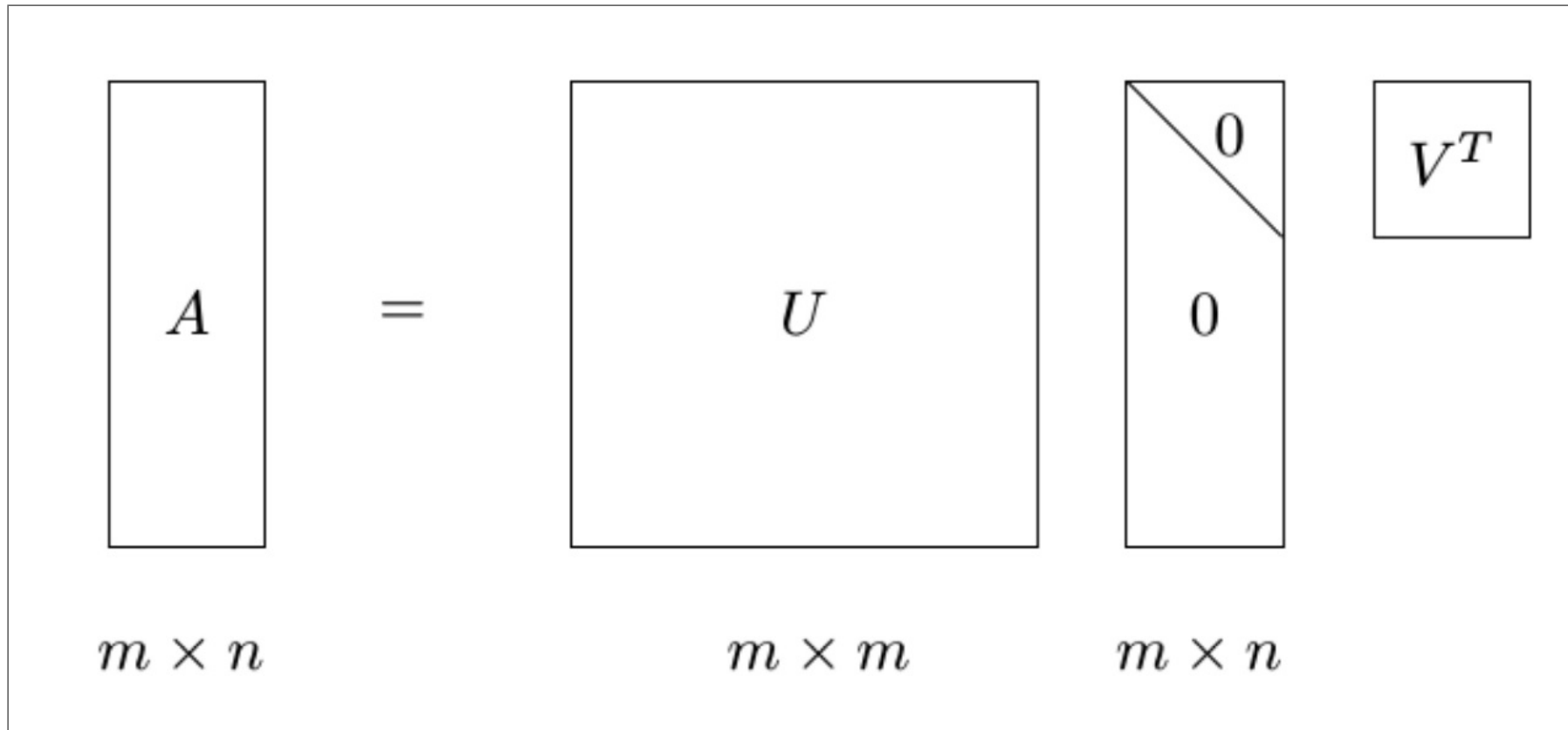
Theorem: Any matrix  $A \in R^{m \times n}$  is diagonal where  $U \in R^{m \times m}$  and  $V \in R^{n \times n}$  are orthogonal, and  $\Sigma \in R^{m \times n}$  is diagonal

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) = \begin{bmatrix} \sigma_1 & 0 & & \\ 0 & \sigma_2 & & \\ & \ddots & \ddots & \\ & & \sigma_n & \end{bmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$$

small sigma is the singular value

The columns of  $U$  and  $V$  are called left singular vectors and *right singular vectors* respectively, and the diagonal elements  $\sigma_i$  are called *singular values*. The SVD is illustrated symbolically in the image down below:



The SVD of a matrix  $A \in \mathbb{R}^{m \times n}$  can be thought of as a weighted, ordered sum of the matrices  $u_i v_i^T$ , where  $u_i = U(:, i)$  and  $v_i = V(:, i)$  are the  $i$ -th orthonormal columns of  $U$  and  $V$

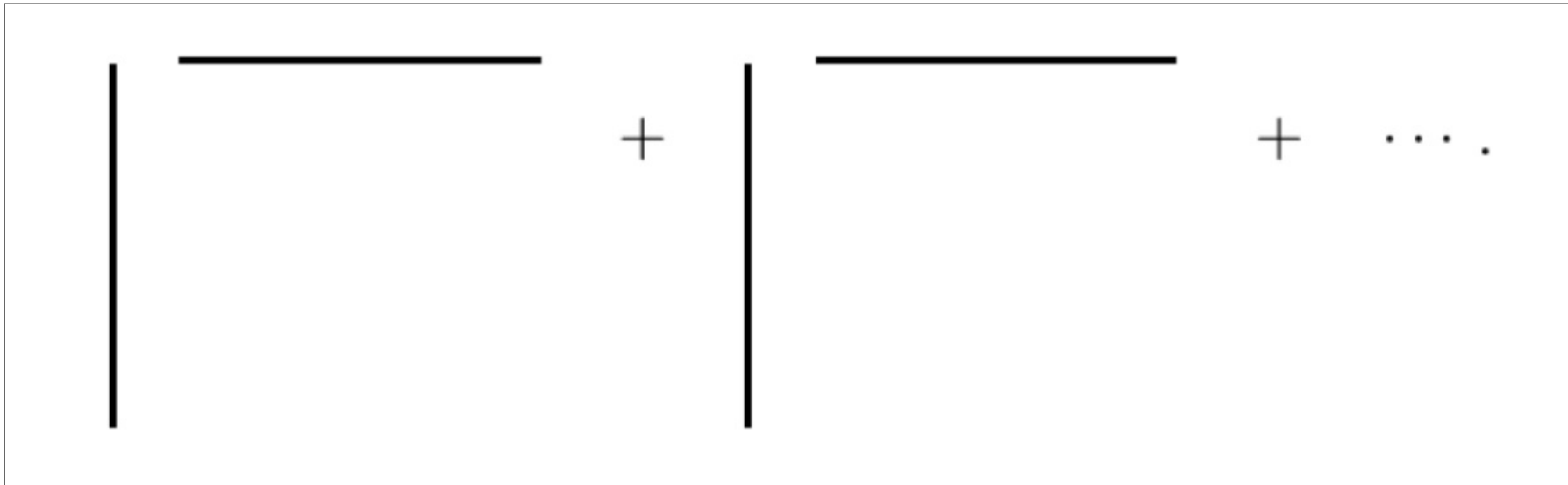
In particular, the matrix  $A$  can be decomposed in the following way:

$$A = \sum_{i=1}^n \sigma_i u_i v_i^T$$

This is usually called the outer product form and it is derived from the thin SVD of  $A$

$$A = U_1 \Sigma_1 V^T = (u_1, u_2, \dots, u_n) \begin{bmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \\ \vdots \\ \sigma_n v_n^T \end{bmatrix} = \sum_{i=1}^n \sigma_i u_i v_i^T$$

The outer product form of the SVD is illustrated symbolically down below:


$$\begin{array}{|} \hline \\ \hline \end{array} \begin{array}{|} \hline \\ \hline \end{array} + \begin{array}{|} \hline \\ \hline \end{array} \begin{array}{|} \hline \\ \hline \end{array} + \dots$$

## PROPERTIES OF SVD

The SVD of a matrix  $A \in \mathcal{R}$  has the following properties:

- the singular values are unique and for distinct positive singular values  $\sigma_i > 0$ , the  $i$ -th columns of  $U$  and  $V$  are also unique up to a sign change of both columns
- $\|A\|_F^2 := \sum_{i,j} a_{ij}^2 = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_p^2 \quad p = \min\{m, n\}$

- Suppose that  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = 0 = \sigma_p$ , then:  
 Rank (A)=r  
 $\text{Range}(A) := \{y | y = Ax \text{ for arbitrary } x\} = \text{span}\{u_1, \dots, u_r\};$   
 $\text{null}(A) := \{x | Ax = 0\} = \text{span}\{v_{r+1}, \dots, v_n\}$
- The singular values of the matrix A are equal to the square root of the eigenvalues  $\lambda_1, \dots, \lambda_m$  of the matrix  $A^T A$
- if A is invertible, then  $A^{-1} = V \Sigma^{-1} U^T$  so that:

$$A^{-1} = \sum_{i=1}^n \frac{1}{\sigma_i} v_i u_i^T$$



Using the SVD, it is possible to define the condition number of a matrix  $A \in R^{n \times n}$  let  $p = \min\{m, n\}$  and  $r = \text{rank}(A)$  if  $p = r$ , the condition number  $k(A)$  is:

$$k(A) = \frac{\sigma_r}{\sigma_1}$$

## THE TRUNCATED SVD

one of the most interesting aspects of the SVD is that enables us to deal with the concept of matrix rank Truncated SVD factorized data matrix where the number of columns is equal to the truncation. It drops the digits after the decimal place for shorting the value of float digits mathematically.

In several applications, the most known one is the compression of data, one may need to determine a low-rank approximation of a matrix  $A$ .

It turns out that the truncated SVD is the solution of approximation problems where one wants to approximate a given matrix by one of lower rank.

we define the trunked SVD like this: Let the SVD of a matrix  $A \in \mathbb{R}^{m \times n}$  be given if

$$k \leq r = \text{rank}(A)$$

and

$$A_k = \sum_{i=1}^K \sigma_i u_i v_i^T$$

then:

$$\min_{\text{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

## THE PROOF:

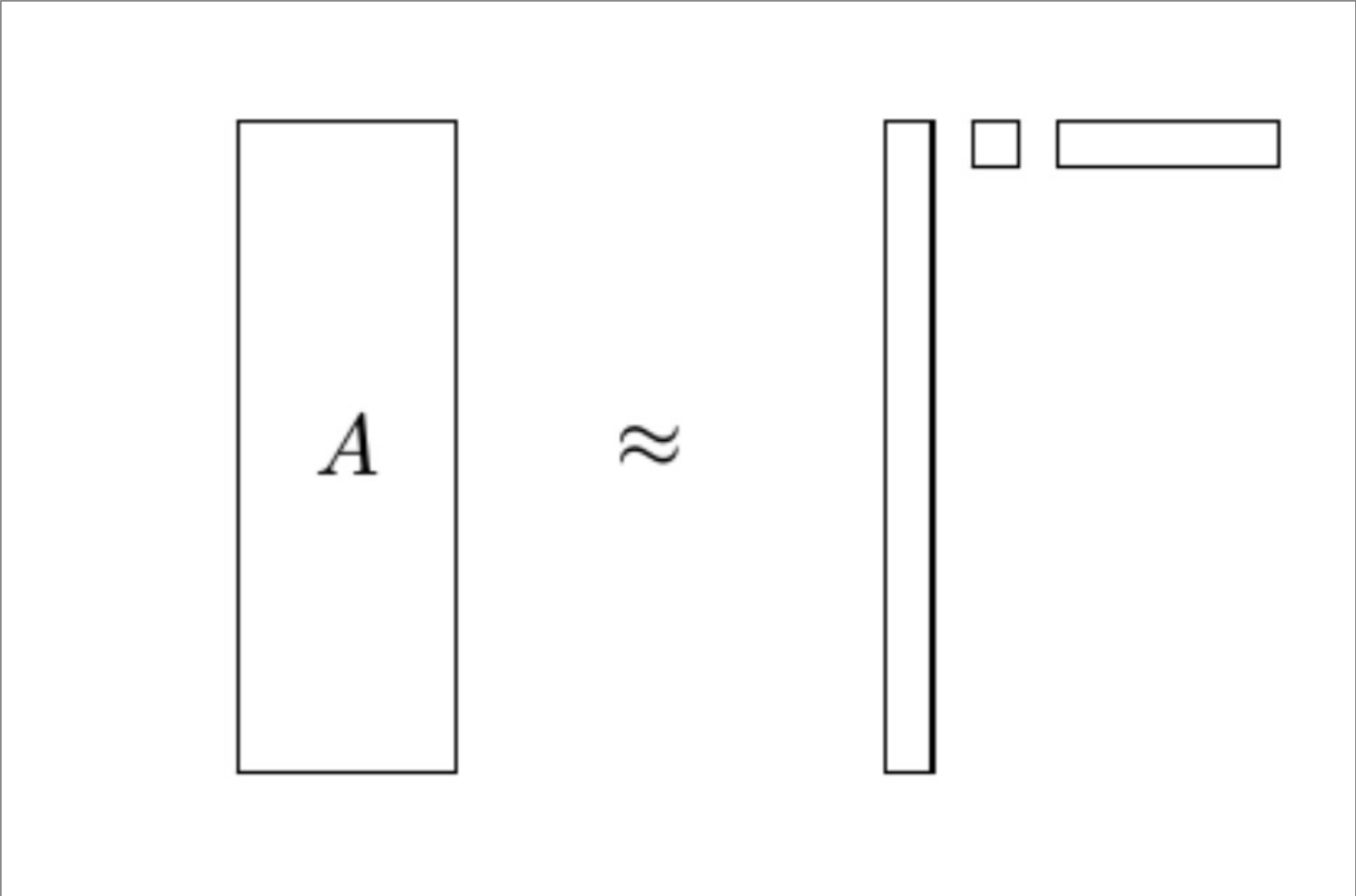
Since  $U^T A_k V = \text{diag}(\sigma_1, \dots, 0, \dots, 0)$   $\text{rank}(A) = k$  and  $U^T (A - A_k) V = \text{diag}(0, \dots, 0, \sigma_{k+1}, \dots, \sigma_p)$ , where:  
 $p = \min\{m, n\}$ . So, since orthogonal matrices preserve the norm,

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

Now suppose  $\text{rank}(B) = k$  for some  $B \in R^{m \times n}$  it follows that we can find orthonormal vectors  $x_1, \dots, x_{n-k}$  such that  $\text{null}(B) = \text{span}\{x_1, \dots, x_{n-k}, v_1, \dots, v_{k+1}\}$  are  $n + 1$  vectors in  $R^n$ ,  $\text{span}\{x_1, \dots, x_{n-k}\} \cap \text{span}\{v_1, \dots, v_{k+1}\} \neq 0$

It is possible to give a similar approximation result with the Frobenius norm

**TRUNCED SVD OR THE LOW-RANK APPROXIMATION OF A MATRIX IS  
ILLUSTRATED SYMBOLICALLY AS:**



## THE HIGHER-ORDER SINGULAR VALUE DECOMPOSITION

Here I present two different Tensor Decomposition techniques:

- HOSVD or the (Higher-Order Singular Value Decomposition) : This is the generalization of the matrix SVD to N-mode tensors
- Tensor-Train Decomposition, which decomposes an N dimensional tensor in a product of 3-dimensional tensors



The HOSVD, also known as Tucker Decomposition, was first introduced by Tucker. It decomposes a tensor into a core tensor multiplies by an orthogonal matrix along each mode  
this can be shown as:

$$\mathcal{A} = \mathcal{S} \times \mathbf{U}^{(1)} \times \mathbf{U}^{(2)} \times \dots \times \mathbf{U}^{(N)}$$

this can visually be seen in the picture down below:

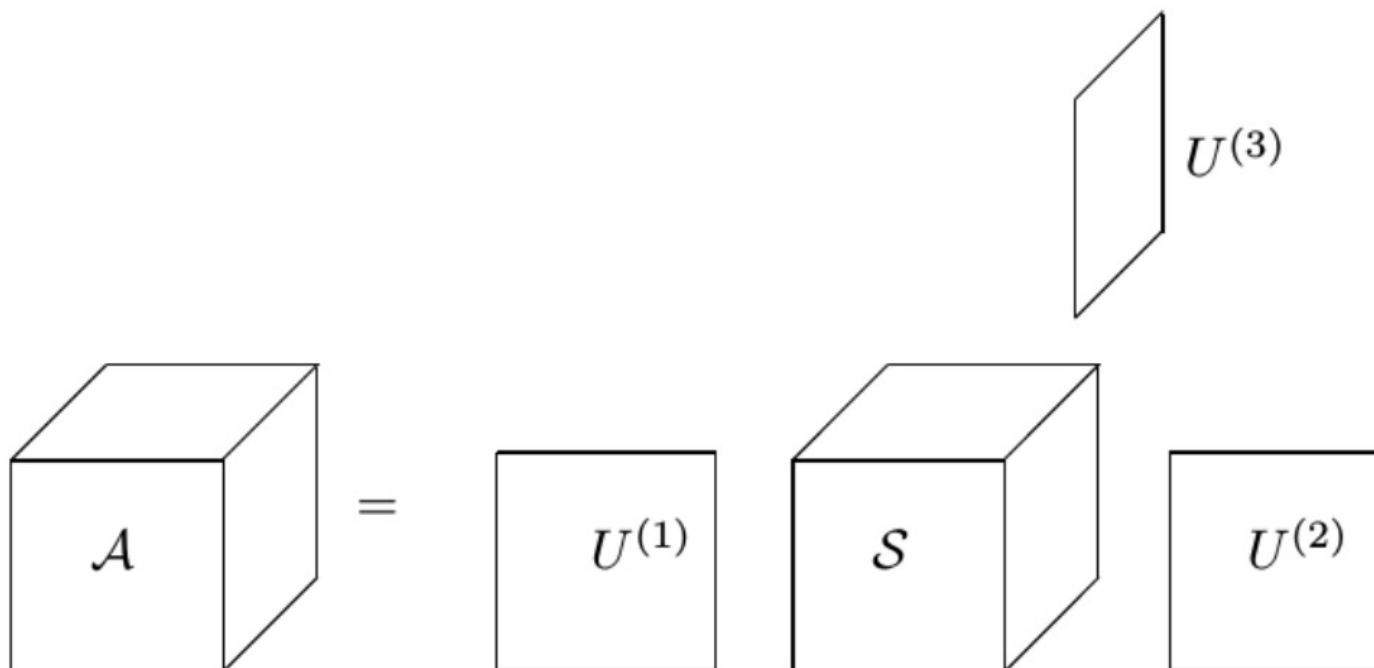


Figure 3.1: Visualization of the HOSVD [11, Figure 8.2].

This means that, if the n-mode singular values are defined as:

$$\sigma_i^{(n)} = \|S_{i_n=i}\|$$

then:

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \geq \sigma_{I_n}^{(n)} \geq 0 \quad \forall n = 1, \dots, N$$

The proof shows the strong connection between the HOSVD of  $A$  and the  $SVD$  of it's matrix unfolding. The derivation is given in terms of real-values tensors.

consider the 1-mode unfolding of  $A$ ,  $A_{(1)}$  and it's SVD

$$A_{(1)} = U^{(1)} \Sigma^{(1)} \left( V^{(1)} \right)^T$$

in which  $\Sigma^{(1)} = \text{diag}(\sigma_1^{(1)}, \sigma_2^{(1)}, \dots, \sigma_{I_1}^{(1)})$

Automatic Face Recognition has become increasingly important in the past few years due to its several applications in daily life such as social media platforms and security services. In this chapter we first describe a Face Recognition algorithm based on the SVD which does not work well with huge databases.

To cope with this problem we introduce two algorithms based on the tensor decompositions seen in the previous chapters: HOSVD and Tensor-Train.

## FACE RECOGNITION USING SVD

Before moving further, we deal with a variant of Algorithm 3, where the truncated SVD instead of the exact SVD is considered. In particular, we truncated according with a parameter  $p$  related to the singular values, whose characteristic pattern is given in Figure

## FACE RECOGNITION WITH TRUNCATED SVD

Before moving further, we deal with a variant of Algorithm 3, where the truncated SVD instead of the exact SVD is considered. In particular, we truncated according with a parameter  $p$  related to the singular values, whose characteristic pattern is given in Figure

In truncated SVD we consider the vector  $w$ , whose entries are given by:

$$w(l) = \frac{\sum_{i=1}^l \sigma_i}{\sum_{i=1}^{n_e} \sigma_i}$$

The, we considered only the first  $k$  singular values, where  $k$  is such that:

$$w(l) \leq p \quad \forall 1 \leq l \leq k \text{ and } w(l) > p \quad \forall k > l$$

it is good to mention that because the error of the truncation is low, if the decay of the singular values is fast enough. Thus, it is worth considering a truncated version of the algorithm, since it gives recognition performance with higher efficiency in terms of speed and memory requirements.



## FACE RECOGNITION USING TENSOR DECOMPOSITION

we used a database of  $n_p$  persons with  $n_e$  expressions which was represented by  $p$  matrices  $A_p \in R^{n_i \times n_e}$  where  $n_i$  is the number of pixels of each image. Now using the multilinear algebra and the algebra of higher-rder tensors, the same dataset can be represented as a tensor  $A \in R^{n_i \times n_e \times n_p}$

## FACE RECOGNITION USING HOSVD

Consider  $A \in \mathbb{R}^{n_i \times n_e \times n_p}$  it is possible to decompose  $A$  using the HOSVD as in:

$$A = Sx_i F x_e G x_p H$$

where  $x_i, x_e, x_p$  are the 1-mode, 2-mode, 3-mode multiplication. To give an interpretation of the HOSVD of  $A$ , it is better to present the decomposition in the form of:

$$A = D x_e G x_p H,$$

where  $D = sx_i F$ , By fixing a particular expression  $e_0$  and a particular person  $p_0$  we would have the vector:

$$A(:, e_0, p_0) = Dx_e g_{e_0} x_p h_{p_0}$$

where  $g_{\{e_0, : \}}$  and  $h_{\{p_0\}} = H(p_0, :)$ , in order words, a particular expression  $e_0$  is characterized by the vector  $g_{e_0}$  and a particular person  $p_0$  is characterized by the vector  $h_{p_0}$  via the bilinear form:

$$Dx_e g x_p h$$

Note:  $x_i$  is the image\_mode multiplication,  $x_e$  is the expression-mode multiplication,  $x_p$  is the person mode multiplication

To develop automatic procedures for face recognition that are robust with respect to varying conditions is a challenging research problem that has been investigated using several different approaches. Principal component analysis (i.e., SVD) is a popular technique that often goes by the name “eigenfaces”

We will discuss how the tensor SVD (HOSVD) can also be used for dimensionality reduction to reduce the flop count.

My code implementation: It is also possible to precompute a QR decomposition of each matrix  $B_e$  to further reduce the work. Thus we arrive at the following algorithm. It is also possible to precompute a QR decomposition of each matrix  $B_e$  to further reduce the work. Thus we arrive at the following algorithm.

Due to the ordering properties of the core, with respect to the different modes (Theorem 8.3), we may be able to truncate the core in such a way that the truncated HOSVD is still a good approximation of  $A$

Therefore, if the rate of decay of the image mode singular values is fast enough, it should be possible to obtain good recognition precision, despite the compression. Thus a substantial rank reduction (from 100 to 10) was possible in this example without sacrificing classification accuracy.

Singular Value Decomposition (SVD) is a useful tool in Functional Data Analysis (FDA). Compared to Principal Component Analysis (PCA), SVD is more fundamental, because SVD simultaneously provides the PCAs in both row and column spaces. Compared to the PCA method, Singular Value Decomposition (SVD) can be thought of as more fundamental, because SVD not only provides a direct approach to calculate the principal components (PCs), but also derives the PCAs in row and column spaces simultane

In the statistical literature, the rows of  $X$  are often viewed as observations for an experiment, and the columns of  $X$  are thought of as the covariates. SVD provides a useful factorization of the data matrix  $X$ , while PCA provides a nearly parallel factoring while **PCA** provides a nearly parallel factoring, via eigen-analysis of the sample covariance matrix, i.e.  $X^T X$ , when  $X$  is column centered at 0



The eigenvalues for  $X^T X$  are then the squares of the singular values for  $X$ , and the eigenvectors for  $X^T X$  are the singular rows for  $X$ .

we extend the usual (column centered) PCA method into a general SVD framework, and consider four types of SVDs based on different centerings.

we categorize the type of SVD and selecting the appropriate type of SVD, including approximation performance, complexity, and interpretability

let  $X = (x_{ij})_{m \times n}$  with  $\text{rank}(x) = r$  the SVD of  $X$  is defined as:

$$X = USV^T = s_1 u_1 v_1^T + (s_2 u_2 v_2^T) \dots + s_r u_r v_r^T$$

where  $U = (u_1, u_2, \dots, u_r)$ ,  $V = (v_1, v_2, \dots, v_r)$ , and  $S = \text{diag} s_1, s_2, \dots, s_r$  with  $s_1 \geq s_2 \geq \dots \geq s_r \geq 0$

The vectors  $u_i$  and  $v_i$  are called singular columns and singular rows respectively. the scalars  $s_i$  are called singular values; and the matrices

$$s_i u_i v_i^T (i = 1, \dots, r)$$

are referred to as SVD components.

Let  $r_i : i = 1, \dots, m, c_j : j = 1, \dots, n$  be the row and column vectors of the matrix  $X$  respectively. The singular columns  $u_i$  form an orthonormal basis for the column space spanned by  $c_j$ , and the singular rows  $v_j$  form an orthonormal basis for the row space spanned by  $r_i$

The SVD factorization has an important approximation property. Let  $A$  be a rank  $k$  ( $k \leq r$ ) (approximation) matrix, and define  $R = X - A = (r_{ij})_{m \times n}$  as it's residual matrix.

## FOUR TYPES OF CENTERING FOR SVD

PCA is very similar to the SVD with the only difference being column mean centering. This might raises a natural question of: "why not do row mean centering?" to answer this we study and compare four possible types of centering and correspondgly four types of SVD.

## FOUR TYPES OF SVD

- no centering (simple SVD or SSvd)
- column centering (column SVD or CSVD)
- row centering (row SVD or RSVD)
- and centering both row and column directions, which is referred ad double centering (Double SVD or DSVD)

let  $\mu$  be the overall mean (or grand mean) of all the elements in  $X$ ,  $\mu_c$  be the column mean vector with the elements being the means of the columns, and  $\mu_r$  be the row mean vector with the elements being the means of the corresponding rows.

We define the sample column mean matrix as  $CM = \mathbf{1}_{m \times 1} \mu_c^T$  sample row mean matrix as  $RM = \mu_r \mathbf{1}_{1 \times n}$  and the sample double mean matrix as  $DM = \mathbf{1}_{1 \times n} \mu_c^T + \mu_r \mathbf{1}_{m \times 1}$

We can use the same formula  $X = A + R$  for all these four types of *SVD*, where  $A$  is the approximation matrix, and  $R$  is the residual matrix.

for no centering,  $A = A^{(s)}$  is the sum of the first several SVD components of  $X$ , which is the best approximation matrix at the corresponding rank. For column centering  $A = CM + A^{(c)}$  where  $A^{(c)}$  is the sum of the first several SVD components of  $X$ -CM.

Similarly for row centering and double centering, we have  $A = RM + A^{(r)}$  and  $A = DM + A^{(d)}$  where  $A^{(r)}$  is the sum of the first several SVD components of  $X$ -RM and  $A^{(d)}$  is the sum of the first several SVD components of  $X$ -DM.

Note that  $A$  here is not the same for the four centering, while we just use the same notation for simplicity. Note that  $CM$  and  $RM$  have rank 1, and  $DM$  is at most rank 2.

Another natural choice of centering is removing the overall mean and then applying SVD. When all the data observations are far away from the origin (the mean is relative larger than the variability of the data set), removing the overall mean will decrease the magnitudes of observations, and thus improve the numerical stability of the SVD.

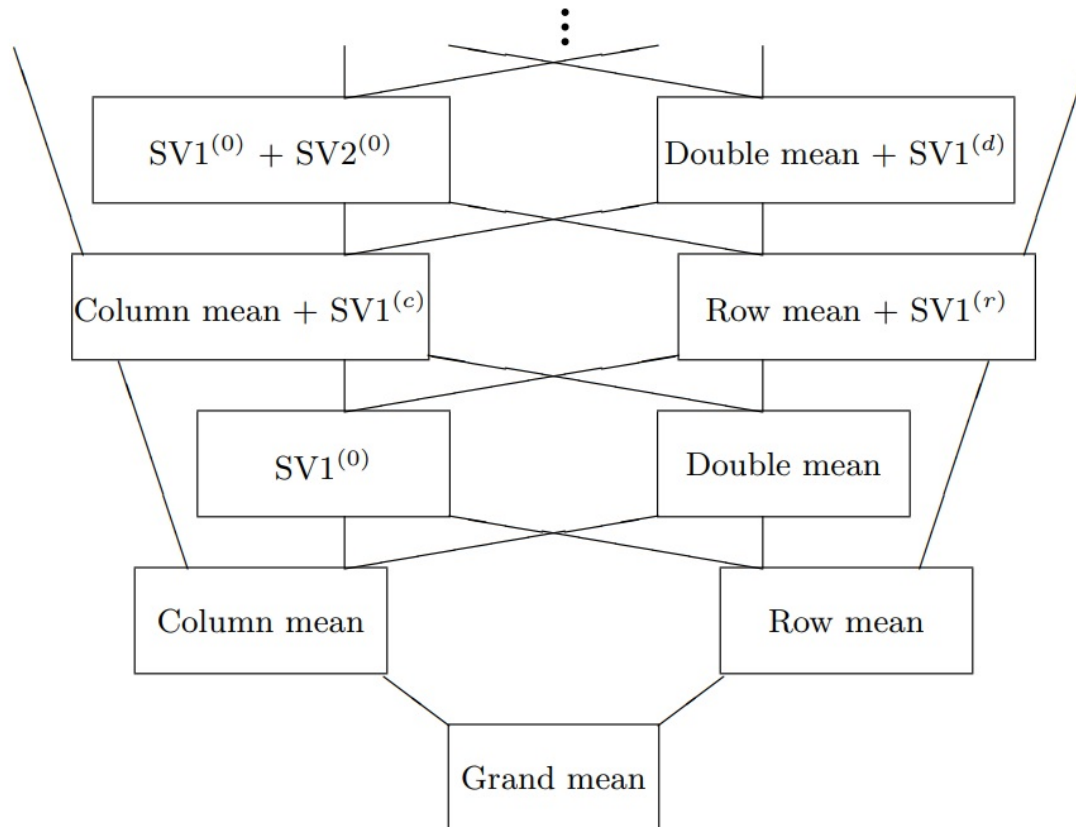
with the functional data set, the overall mean does not provide informative curve information of the data set. There are cases where removing the overall mean will cause the data result in losing some good features. For example, the orthogonality of the curves will be lost.

Note, that the column/row/double centerings automatically remove the grand mean. Thus, we will not discuss the case of just removing the overall mean. However, our programs do provide the option of applying SVD after just removing the overall mean.



# THE APPROXIMATION DIAGRAM

## 3.2.1 Approximation Diagram



This is the Relationship between approximations using the four types of SVD. A lower level approximation is always worse than an upper level one. Note that the lower

level also provides a simpler model than the upper level.

For real applications, it is not enough to use only the generalized scree plot to select appropriate models. However, the generalized scree plot is still useful to give an initial impression of which model might be a better candidate. (In multivariate statistics, a scree plot is a line plot of the eigenvalues of factors or principal components in an analysis)

in this research several simulated data sets were designed to illustrate the above idea of model selection. They also showed that each of the four types of centerings can be the most appropriate model.

All these simulated data sets are designed as  $49 \times 48$  the same as the network traffic data set.

In this setting in the first and the third example, each row can be viewed as one daily usage profile, and each column can be treated as a cross-day times series of one specific time in a day. These data sets are designed to have clearly weekly patterns.

The first example is used to illustrate a situation where CSVD gives the best approximation performance.

important: This example is designed to be the sum of a column mean matrix (in this equation it is the  $\mu_c(j)$ )

A multiplicative component (for example, in the equation down below it is,  $f_2(i)g_2(j)$ ) and some noise, thus the model is:

$$h_1(i, j) = \mu_c(j) + f_1(i)g_1(j) + \epsilon(i, j)$$

Here,  $\mu_c(j) = \sin(j\pi/24)$ ,  $g_1(j) = -\cos(j\pi/24)$ ,  $f_1(i) = 1$  when  $mode(i, 7) \neq 0$  and  $6$  or  $f_1(i) = 2$  when otherwise.

Note that we will use the same notation  $\epsilon$  for different realizations of all the simulated examples. also note that In terms of network usages, the weekdays and weekends do not have the same usage magnitudes (due to the multiplicative component), nor the same usage patterns (because of the column mean component)

The generalized scree plot suggests that there are 2 components for the CSVD/DSVD/SSVD models, while there are 3 components for the RSVD model. The RSVD is the worst model among the four, because the row mean matrix explains a very low proportion of the TSS. The scree plot shows that CSVD with 2 components is the leftmost one, which suggests the CSVD is the best model among the four types of centerings, in terms of complexity and approximation performance.

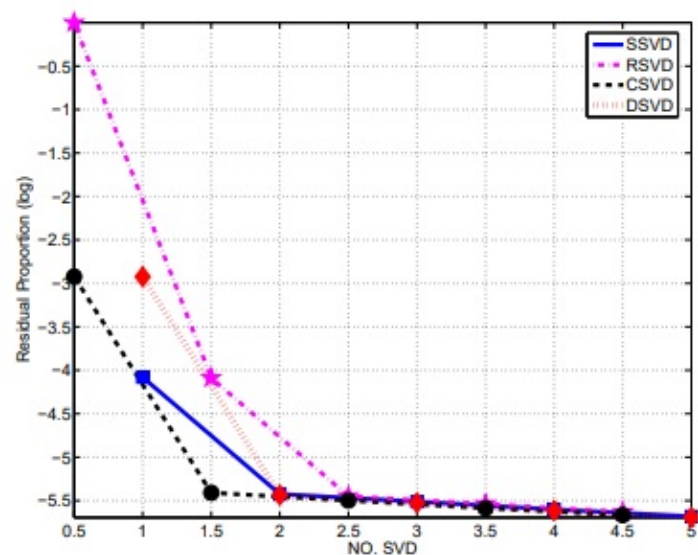


Figure 6: *Log-scale generalized scree plot for the first toy example. By using the scree rule mentioned in the text, one is suggested that there are two components for CSVD/SSVD/DSVD, and three components for RSVD. CSVD is the leftmost one, thus it is the appropriate model for the first toy example.*



By examining the surface plots of all four centerings, we find the RSVD and SSVD provide similar decomposition, except, the RSVD has one row mean matrix. This similarity is due to the low proportion of the row mean matrix, as discussed earlier. The surface plots of the CSVD (The left four panels in Figure 7) show the common usage pattern in the second panel of the first row

The first SVD component of the CSVD shows the contrast between the weekdays and weekends. It shows that after removing a common daily usage profile, the contrast curves between the weekdays and weekends have the same shape, but have different contrast magnitudes.

the surface plots for the SSVD (the right four panels in Figure 7) use two components for the major variation of the data matrix. If the daily usages share the same usage pattern but with different magnitudes, one SSVD component should be enough for the major modes of variation. Thus, the two SSVD components for this example suggest that weekdays and weekends do have different usage patterns and different magnitudes

This suggests that the SSVD model for this example has better interpretation than the CSVD model.

For the model selection of (5), we find that it is hard to distinguish whether SSVD or CSVD is “better”. For the two good candidates, the CSVD is a simpler model (mean plus one multiplicative CSVD component is simpler than two multiplicative SVD components), while the SSVD has better interpretation (weekdays and weekends do not share the same usage pattern, nor do the usage magnitudes).

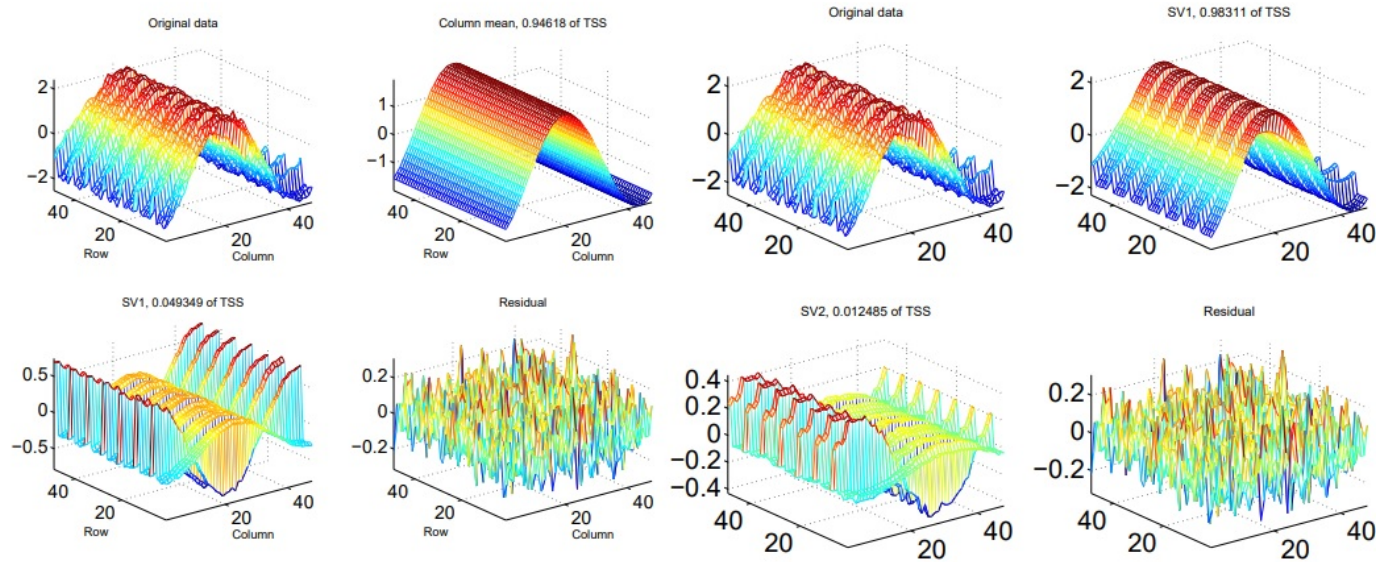


Figure 7: The left four panels are the surface plots of CSVD for the first toy example, and the right four panels are the surface plot of SSVD for the first toy example, from equation (5). For the first toy example, RSVD and DSVD are worse in terms of approximation. However, SSVD and the CSVD are candidates of good approximation performance. CSVD is better than SSVD, because it contains simpler model. However, SSVD shows that weekdays and weekends have different daily shapes and different magnitudes of daily

related to the previous slide: This suggests that the SSVD model for this example has better interpretation than the CSVD model.

## THE SECOND EXAMPLE

The second example is designed to illustrate that RSVD has the best approximation performance among the four in a context that is much different from just the transpose of Example 1. The model is set up as the sum of a row mean matrix ( $f_2(i)$  in equation (6)), a multiplicative component ( $f_3(i)g_3(j)$ ) in equation (6)) and noise, and can be written mathematically as:

$$h_2(i, j) = f_2(i) + f_3(i)g_3(j) + \epsilon(i, j)$$

in this formula: where  $f_2(i) = \cos(\frac{i\pi}{24})$ ,  $f_3(i) = \sin(\frac{i\pi}{24})$ ,  $g_3(j) = 1$  when  $1 \leq j \leq 12$

The model of the second simulated example is different from the first one in an important way. The function  $g_3(j)$  in the multiplicative component is orthogonal to the constant vector  $\mathbf{1}_{48 \times 1}$  and  $f_3(i)$  is orthogonal to  $f_2(i)$

These make the row mean matrix  $f_2(i)$  and the multiplicative component  $f_3(i)g_3(j)$  orthogonal to each other in both the column space and the row space. In this example, the rows of the simulated data are not useful models for daily network traffic.

## THE IMAGE DOWN BELOW ILLUSTRATE THIS VIVIDLY:

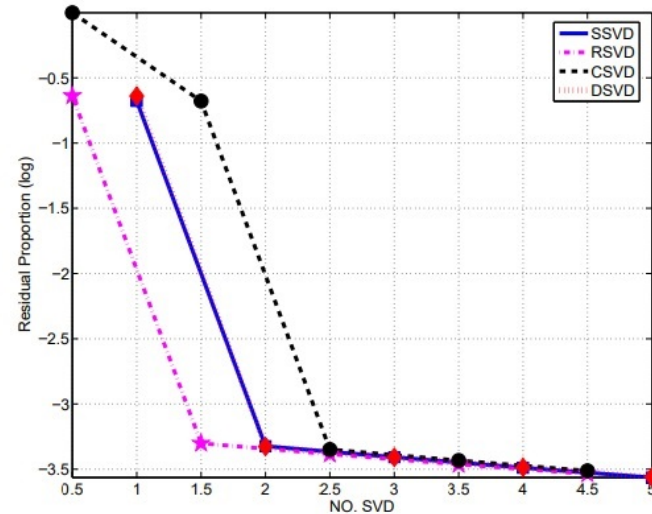


Figure 8: *Log-scale generalized scree plot for the second toy example. By using the scree rule mentioned in the text, one is suggested that there are two components for DSVD/RSVD/SSVD, and three components for CSVD. RSVD is the leftmost one, thus it is the appropriate model for the second toy example in terms of approximation performance and complexity.*



The generalized scree plot shows the CSVD is the worst model, because of the low percentage of TSS explained by the column mean matrix. It also suggests that the RSVD and the SSVD are better models. Based on the rules discussed in Section 3.3, the generalized scree plot suggests the RSVD will be the “optimal” model among the four centerings

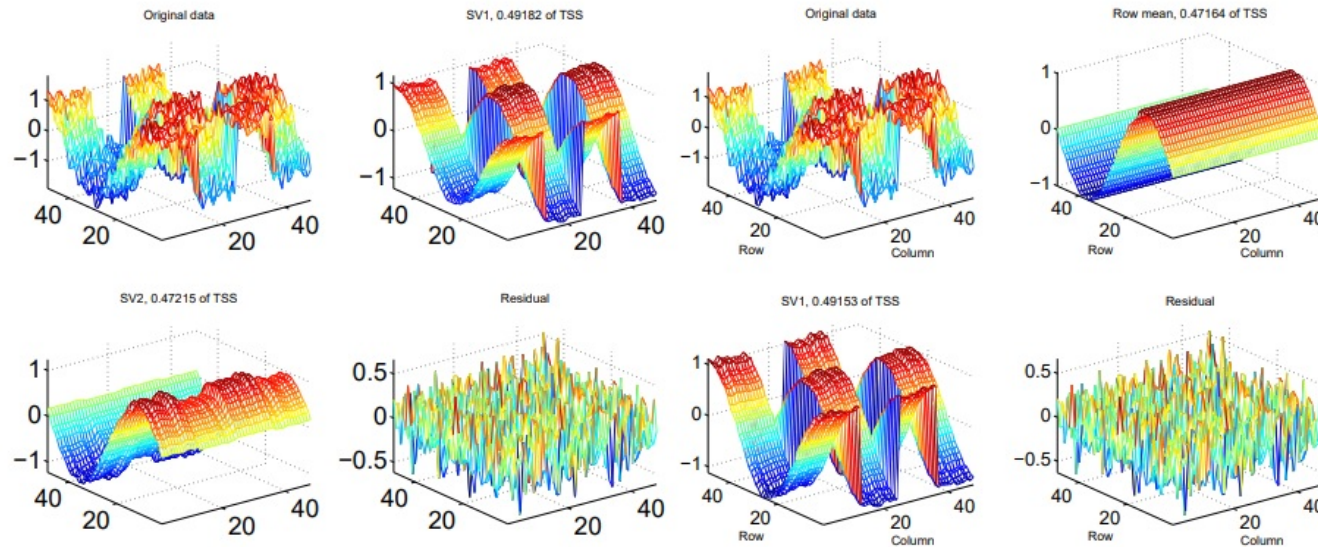


Figure 9: The left four panels are the surface plots of the SSVD for the second toy example, and the right four panels show the surface plots of the corresponding RSVD model. The SSVD picks the multiplicative component in (6) as the first component. And the second SSVD component is close to the row mean matrix. The reason for this is because these two components are orthogonal to each other in both the row and the column space, and the multiplicative component has larger variations than the row mean matrix. The RSVD gives an opposite decomposition, which is almost the same as the model where the data matrix is generated from.

The SSVD surface plots are the left four panels in Figure 9, and the right four panels in Figure 9 show the RSVD surface plot of the simulated data set. By looking at the surface plots, we find that the SSVD essentially picks the second part  $f_3(i)g_3(j)$  as the first SVD component, and the first part  $f_2(i)$  as the second component.

## REFERENCES

- Book: Matrix Methods in Data Mining and Pattern Recognition
- Paper: Singular Value Decomposition and Its Visualization by Lingsong Zhang\* , J. S. Marron, Haipeng Shen and Zhengyuan Zhu
- Tensor decompositions for Face Recognition by Domitilla Brandoni

# Thank you for your Attenstion

Speaker notes

Speaker notes

Speaker notes