Dimensionality reduction

JISHNU M
Student id: 22020503

October 2022

1 Introduction

While dealing with input data for a predictive model, we often presume that if we consider more features, then we get more accuracy for our model. But this may not be true always. More features not only make our model complex but can even mislead our predictions. So choosing relevant features is very important in Machine Learning. Here comes the importance of dimensionality reduction.

Dimensionality reduction refers to the method of decreasing the number of input variables for a predictive model so that we get optimum accuracy for the model with minimal features. Lesser input variables can result in a uncomplicated predictive model which may sometimes shows better performance.

Two popular dimensionality reduction methods are:

1. Singular Vector Decomposition (SVD)
2. Linear Descriminant Analysis (LDA)

2 Singular Vector Decomposition (SVD)

SVD is a method to factorize a rectangular/square matrix A into 3 matrices.

$$A = U\Sigma V^T \tag{1}$$

where

- A is a $m * n$ matrix
- U is an orthogonal $m * k$ matrix
- Σ is a $k * k$ diagonal matrix
- V^T is an orthogonal $k * n$ matrix

ie,

$$A = \begin{bmatrix} u_1 & u_2 & . \\ . & . & . \\ . & . & . \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & . \\ 0 & \sigma_2 & . \\ . & . & . \end{bmatrix} \begin{bmatrix} v_1 & . & . \\ v_2 & . & . \\ . & . & . \end{bmatrix} \tag{2}$$

The vectors $\begin{bmatrix} u_1 \\ . \end{bmatrix}$, $\begin{bmatrix} u_2 \\ . \end{bmatrix}$ are called left singular vectors and σ_1, σ_2 are called singular values and the vectors $\begin{bmatrix} v_1 & . \end{bmatrix}$, $\begin{bmatrix} v_2 & . \end{bmatrix}$ are called right singular vectors.

2.0.1 Σ-Singular value matrix

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & . & . & \\ 0 & \sigma_2 & . & . & \\ . & & . & & \\ . & & & . & \\ & & & & \sigma_k \end{bmatrix}$$

is a $k * k$ diagonal matrix with $\sigma_1 > \sigma_2 > .. > \sigma_k$

- Singular values are listed in descending order in the Singular matrix Σ.
- Highest order dimension captures the most variance in the original data set or most of the information related to the term-document matrix.
- The next higher dimension captures the next higher variance in the original data set.
- Larger singular values reflect the major associative patterns in the data and we can neglect smaller singular values having least significance.

2.1 Mathematical concepts for SVD

2.1.1 How to find U Σ and V^T from A

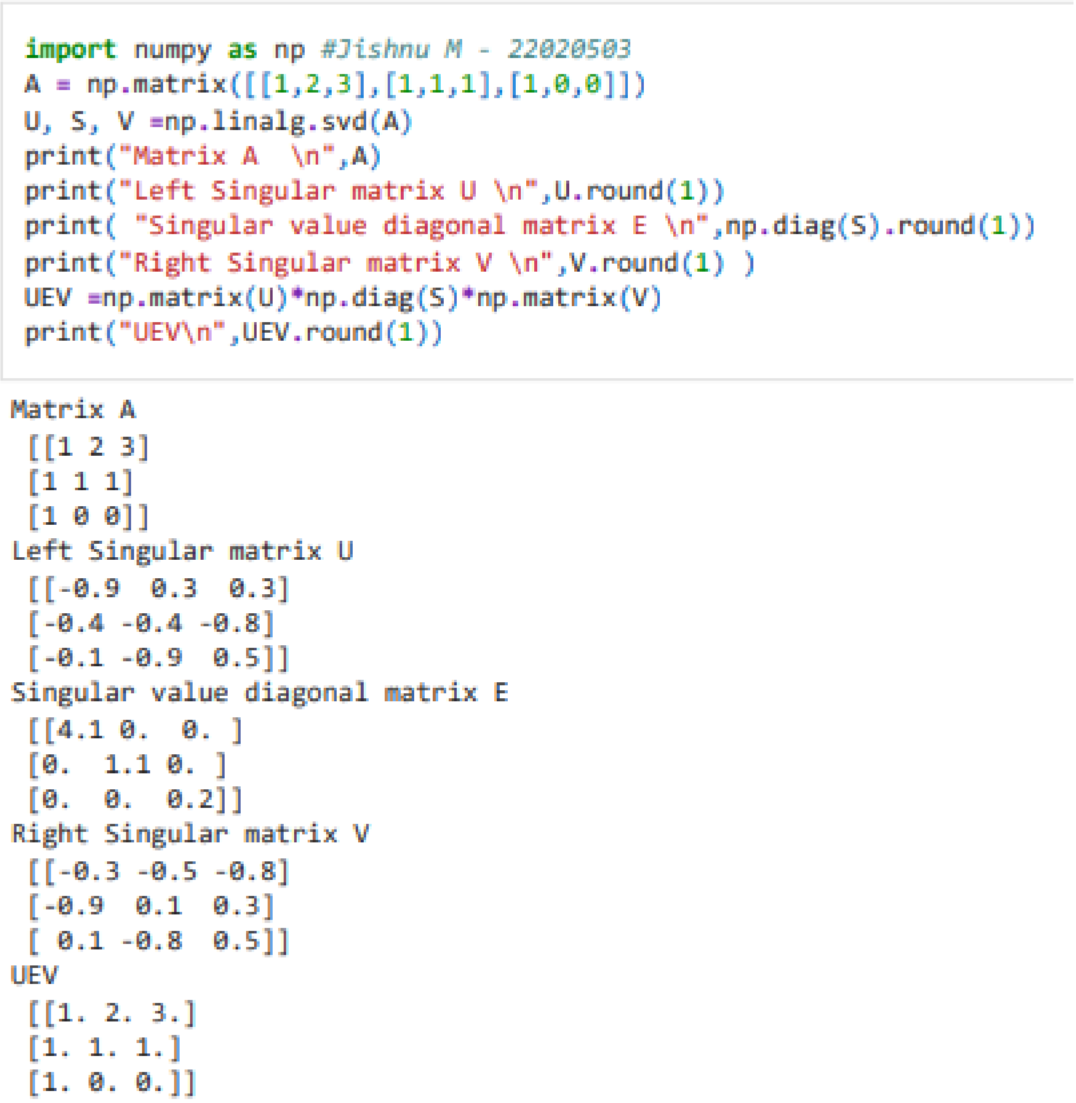
Consider $A^T A$
from (1) we have

$$\begin{aligned} A^T A &= (V \Sigma^T U^T) U \Sigma V^T \\ &= V(\Sigma^T \Sigma) V^T \end{aligned} \tag{3}$$

Here (3) is the usual diagnolization of $A^T A$. Moreover, $A^T A$ will be positive semi-definite symmetric matrix implies that it's eigen vectors will be orthogonal. Hence $(\Sigma^T \Sigma)$ forms eigen value diagonal matrix of $A^T A$. i.e, the σ^2 for A.
Also V will be the eigen vector matrix of $A^T A$.
Similarly if we consider AA^T we will get U as the eigen vector matrix of AA^T.

2.2 Example

Consider $3*3$ matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$



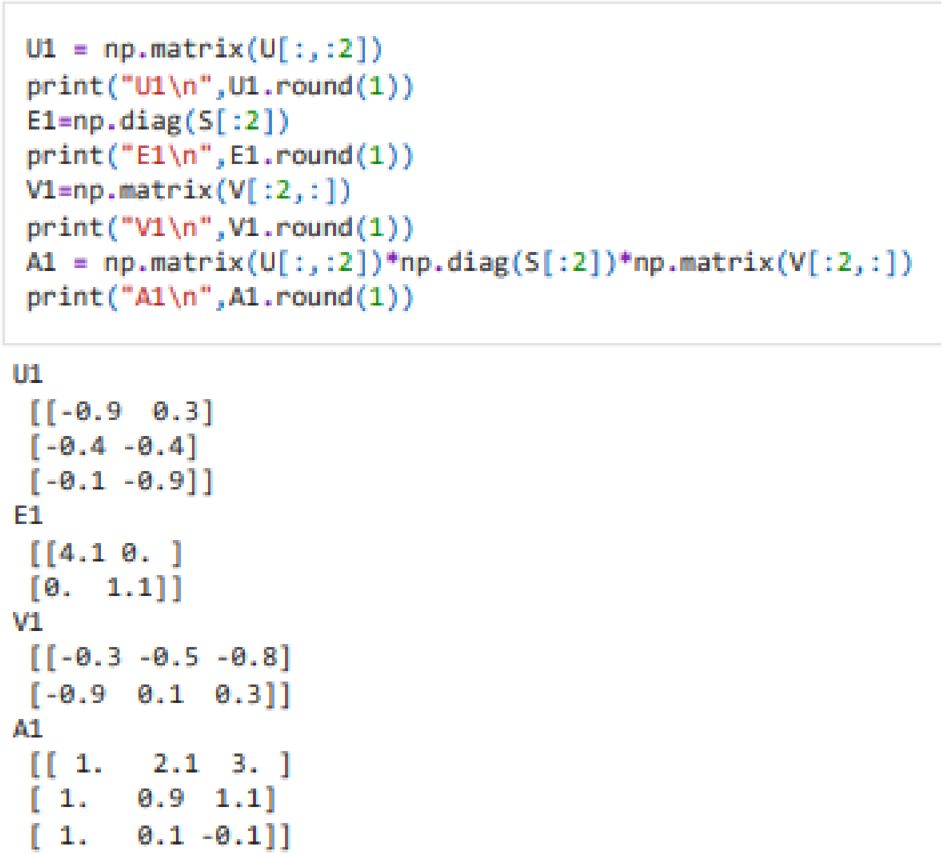
```
import numpy as np #Jishnu M - 22020503
A = np.matrix([[1,2,3],[1,1,1],[1,0,0]])
U, S, V =np.linalg.svd(A)
print("Matrix A  \n",A)
print("Left Singular matrix U \n",U.round(1))
print( "Singular value diagonal matrix E \n",np.diag(S).round(1))
print("Right Singular matrix V \n",V.round(1) )
UEV =np.matrix(U)*np.diag(S)*np.matrix(V)
print("UEV\n",UEV.round(1))
```

```
Matrix A
 [[1 2 3]
 [1 1 1]
 [1 0 0]]
Left Singular matrix U
 [[-0.9  0.3  0.3]
 [-0.4 -0.4 -0.8]
 [-0.1 -0.9  0.5]]
Singular value diagonal matrix E
 [[4.1 0.  0. ]
 [0.  1.1 0. ]
 [0.  0.  0.2]]
Right Singular matrix V
 [[-0.3 -0.5 -0.8]
 [-0.9  0.1  0.3]
 [ 0.1 -0.8  0.5]]
UEV
 [[1. 2. 3.]
 [1. 1. 1.]
 [1. 0. 0.]]
```

Figure 1: Decomposing matrix A as $U\Sigma V^T$

Consider figure 1 in which we have decomposed matrix A as $U\Sigma V^T$. Here U, Σ and V are $3*3$ matrices. Also singular values are $\sigma_1 = 4.1$,$\sigma_2 =$

1.1,$\sigma_3 = 0.2$. Here the singular value σ_3 has least significance, so we can approximate σ_3 and it's corresponding eigen vectors to zero. Hence we can reduce U as $3 * 2$, Σ as $2 * 2$ and V as $2 * 3$ matrices as shown in fig 2 and with minor errors approximation errors we obtain A.



```
U1 = np.matrix(U[:,:2])
print("U1\n",U1.round(1))
E1=np.diag(S[:2])
print("E1\n",E1.round(1))
V1=np.matrix(V[:2,:])
print("V1\n",V1.round(1))
A1 = np.matrix(U[:,:2])*np.diag(S[:2])*np.matrix(V[:2,:])
print("A1\n",A1.round(1))
```

```
U1
 [[-0.9  0.3]
 [-0.4 -0.4]
 [-0.1 -0.9]]
E1
 [[4.1 0. ]
 [0.  1.1]]
V1
 [[-0.3 -0.5 -0.8]
 [-0.9  0.1  0.3]]
A1
 [[ 1.   2.1  3. ]
 [ 1.   0.9  1.1]
 [ 1.   0.1 -0.1]]
```

Figure 2: Decomposing matrix A as $U\Sigma V^T$ with lesser dimensions

In real life application this matrix A can be a data set, and each columns represent different features (in this case 3 features). These 3 features can be reduced to 2 as we have done in the case of matrix A. Like this we can do dimensionality reduction using SVD.

3 Linear Discriminant Analysis LDA

LDA is a classification algorithm which can be also used as a dimensionality reduction algorithm. Consider a classification problem to distinguish two or more classes with numerous features, the Linear Discriminant Analysis model is one of the best method to solve such classification problems.

3.1 Example

Let's take a 2-D dataset

$$C_1 \rightarrow X_1 = (X_1, X_2) = \{(4,1),(2,4),(2,3),(3,6),(4,4)\}$$

$$C_2 \rightarrow X_1 = (X_1, X_2) = \{(9,10),(6,8),(9,5),(8,7),(10,8)\}$$

STEP1:

Compute within class scatter matrix(s$_W$)

$S_w = S_1 + S_2$

S_1 = co variance matrix of class $_1$

S_2 = co variance matrix of class $_2$

$S_1 = \sum_{x \in C_1} (x - \mu_1)(x - \mu_2)^T$

μ_1 = Mean class of C$_1$

x = $DatapresentinC_1$

$$\mu_1 = \left\{ \frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5} \right\}$$

$\mu_1 = [3.00, 3.60]$

similarly,

$\mu_2\bar{[}$8.2 ,7.60]

Mean reduced data,

$$[x_1 - \mu_1] \ = \left[\begin{matrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{matrix} \right.$$

Now for each x we are going to calculate,

$$(x - \mu_1)(x - \mu_1)^T$$

so we will have 5 such matrices.

1) $\begin{bmatrix} 1 \\ -2.6 \end{bmatrix} * \begin{bmatrix} 1 & -2.6 \end{bmatrix} = \begin{bmatrix} 1 & -2.6 \\ -2.6 & 6.76 \end{bmatrix}$

2) $\begin{bmatrix} -1 \\ 0.4 \end{bmatrix} * \begin{bmatrix} -1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & -0.4 \\ -0.4 & 0.16 \end{bmatrix}$

3) $\begin{bmatrix} -1 \\ 0.6 \end{bmatrix} * \begin{bmatrix} -1 & -0.6 \end{bmatrix} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 0.36 \end{bmatrix}$

4) $\begin{bmatrix} 0 \\ 2.4 \end{bmatrix} * \begin{bmatrix} 0 & -2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 5.76 \end{bmatrix}$

5) $\begin{bmatrix} 1 \\ 0.4 \end{bmatrix} * \begin{bmatrix} 1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 0.16 \end{bmatrix}$

Adding these equations and taking average get co variance of S_1

$S_1 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$

Similarly for the class 2 the co variance matrix is given by,
$S_2 = \begin{bmatrix} 2.6 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$

$S_w = \begin{bmatrix} 2.6 & -0.04 \\ -0.44 & 5.28 \end{bmatrix}$

STEP2:

Computing between class scatter matrix

$S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

$= \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16.0 \end{bmatrix}$

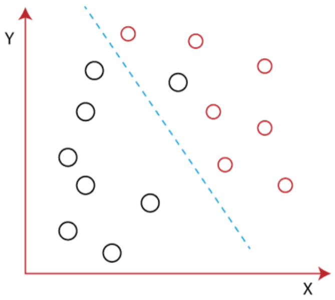


Figure 3: Binary classification problem and class seperating hyperplane

STEP3:

Find the best LDA projection vector similar to principal component analysis.we find this using eigen vector having largest eigen value.

$$S_w^{-1} * S_b V = \lambda V$$

$\begin{bmatrix} S_W^{-1} & S_b - \lambda I & \end{bmatrix}$=$\begin{bmatrix} 11.89-\lambda & 8.81 & \\ 5.08 & 3.76-\lambda & \end{bmatrix}$=0

solving we get $\lambda = 15.65$

$substituting\lambda$in equation we get,

$\begin{bmatrix} V1 \\ V2 \end{bmatrix} = \begin{bmatrix} 0.91 \\ 0.34 \end{bmatrix}$

we get directly solve, $\begin{bmatrix} V1 \\ V2 \end{bmatrix} = \mathrm{S}_w^{-1}(\mu_1 - \mu_2)$

$S_w^{-1} = \begin{bmatrix} 0.384 & 0.032 \\ 0.032 & 0.192 \end{bmatrix}$

$STEP4:$

Dimension reduction

$Y = W^T X \rightarrow (Input)$

- The main objective of the LDA is to find the best fitting hyper plane which shows maximum class seperability as shown in the fig 3.

- LDA maximizes the distance between means of two classes.

- LDA minimizes the variance within the individual class.

- The hyperplane can be found by using the equation $W^T X + b$.

- The cost function corresponding to LDA is $J(W) = \frac{(m_0 - m_1)^2}{\sigma_0^2 + \sigma_1^2}$ where m_0,m_1 corresponds means of the 2 classes and σ_0,σ_1 corresponds the variance of the 2 classes.

3.2 Advantages of LDA

The major advantages of Linear Discriminant Analysis are:

1. Logistic regression is an efficient binary classification algorithm but it fails in case of multiple classification problems with well-seperated classes.

2. LDA reduces computing cost significantly by reducing number of features.

3. LDA is an effective face detection algorithm when it is coupled with eigenfaces.