Homework 4

Arnab Bachhar

September 26, 2023

0.1 Singular Value Decomposition: Problem 1

Solve the following linear algebraic equations by using the singular value decomposition (SVD) method. For the matrix diagonalization, you can use any external diagonalization program. Specify which program you have used. The following items must be discussed in the report: (i) the column-orthogonal matrix U , the square diagonal matrix W , and the orthogonal matrix V . (ii) Show that $A = UWV^T$. (iii) Write down the solution from your code and the general solution for the equations. (iv) Confirm that your solution satisfies the original set of the equations

2x + 3y +10z - u =1 ....(1)

10x + 15y + 3z + 7u =2............................(2)

-4x + y + 2z + 9u =3 ....(3)

Ans:

SVD Decomposition: It is a very essential matrix factorization technique. SVD decomposes A into three matrices:

- U Matrix: The left singular matrix (U) is an $m \times m$ orthogonal matrix (i.e., $U^TU = I$), where each column represents the left singular vectors of A.
- Σ Matrix: The middle matrix (Σ) is an $m \times n$ diagonal matrix. The diagonal elements, called the "singular values," are arranged in descending order.
- V^T Matrix: The right singular matrix (V^T) is an $n \times n$ orthogonal matrix, where each row represents the right singular vectors of A.

Algorithm 1 Custom Singular Value Decomposition (SVD)

Require: Matrix A, Tolerance tol

Ensure: Matrices U, w, V^T (SVD decomposition)

Step 1: Compute the covariance matrix

$ATA \leftarrow A^T \cdot A$

Step 2: Compute eigenvalues and eigenvectors of ATA

$eigenvalues, eigenvectors \leftarrow \text{eigen}(ATA)$

Step 3: Sort eigenvalues and eigenvectors in descending order

$sorted\_indices \leftarrow \text{sortperm}(eigenvalues, \text{rev} = true)$

$eigenvalues \leftarrow eigenvalues[sorted\_indices]$

$eigenvectors \leftarrow eigenvectors[:, sorted\_indices]$

Step 4: Calculate the matrix V

$V \leftarrow eigenvectors$

$U \leftarrow \text{zeros}(\text{size}(A, 2), \text{size}(A, 2))$

Step 5: Calculate the matrix U

for $i = 1$ **to** $\text{length}(eigenvalues)$ **do**

  $U[:, i] \leftarrow \frac{A \cdot V[:, i]}{\|A \cdot V[:, i]\|}$

end for

Step 6: Ensure that U**,** V **are real**

$U \leftarrow \text{real}(U)$

$V \leftarrow \text{real}(V)$

Step 7: Handle the case when singular values are close to zero

$non\_zero\_singular\_values \leftarrow singular\_values > tol$

$k \leftarrow \text{sum}(non\_zero\_singular\_values)$

$singular\_values \leftarrow singular\_values[1 : k]$

$w \leftarrow \text{zeros}(k, k)$

for $i = 1$ **to** $\text{length}(singular\_values)$ **do**

  $w[i, i] \leftarrow singular\_values[i]$

end for

Return U, w, V^T

The custom svd algorithm has been implemented in julia and the pseudocode or algorithm is included in the report. For the first problem , the number of unknown variables are more than the number of equations available; so a dummy equation is taken consisting all zero for the unknown variables and also for right hand side vector.

The output of the code is mentioned below. I have used eigen built-in function in linear algebra package of julia.

$$U = \begin{bmatrix} -0.280122 & -0.507382 & -0.814921 & -0.639602 \\ -0.949887 & 0.023819 & 0.311685 & -0.639602 \\ -0.138733 & 0.861392 & -0.488627 & 0.426401 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$W = \begin{bmatrix} 20.3808 & 0.0 & 0.0 & 0.0 \\ 0.0 & 9.98411 & 0.0 & 0.0 \\ 0.0 & 0.0 & 9.16188 & 0.0 \\ 0.0 & 0.0 & 0.0 & 5.94003e-8 \end{bmatrix}$$

$$\text{Singular values} = [20.380812324036295, 9.98411365816233, 9.161875543430254, 5.9400284832902135e-8]$$

$$V^T = \begin{bmatrix} -0.46633 & -0.747144 & -0.290879 & -0.373767 \\ -0.422886 & -0.0303951 & -0.328479 & 0.844005 \\ 0.375635 & 0.190124 & -0.894075 & -0.152909 \\ 0.680149 & -0.636164 & 0.0901143 & 0.352948 \end{bmatrix}$$

$$\text{Reconstructed } A = \begin{bmatrix} 2.0 & 3.0 & 10.0 & -1.0 \\ 10.0 & 15.0 & 3.0 & 7.0 \\ -4.0 & 1.0 & 2.0 & 9.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Reconstruction Error $= 5.940027399478626e-8$

$$\text{Inverse of } W = \begin{bmatrix} 0.0490658 & 0 & 0 & 0 \\ 0 & 0.100159 & 0 & 0 \\ 0 & 0 & 0.109148 & 0 \\ 0 & 0 & 0 & 0.0 \end{bmatrix}$$

Null basis :

$$=[-0.373767301738, 0.844005216013, -0.152908594610, 0.3529478165215]$$

General Solution :

$$=[-0.10038344269892, 0.05848081921965, 0.1281447617370, 0.256605407758]$$

Residual $(Ax_p - b) = 1.0175362097255202e-15$

Particular Solution : (x_p)

$$=[-0.09853585329530, 0.05430877129987, 0.12890061249715, 0.25486073228051]$$

I have confirmed the solution satisfies the set of equations and also verified the particular solution using built in svd function in linear algebra package in julia.

The computation of the svd function took 2.710306 seconds (4.44 M allocations), consuming 298.203 MiB of memory. There was 5.05% garbage collection time, and compilation accounted for 99.99% of the total time.

0.2 Singular Value Decomposition: Problem 2

Solve the following linear algebraic equations by using the SVD program you coded for Prob.1. Specify which program you have used. The following items must be discussed in the report: (i) the column-orthogonal matrix U , the square diagonal matrix W , and the orthogonal matrix V . (ii) Show that $A = UWV^T$. (iii) Write down the solution from your code. (iv) Confirm that your solution satisfies the original set of the equations. Compare the left-hand side with the right-hand side of the equations. (15 pts)

2x + 3y + u = 1.............. (4)

4x + 6y + 1.99999995u = 2.................. (5)

-4x + y + 2z -2u = 3..................... (6)

-2x -3y + 5z + u = 4................ (7)

Answer: The algorithm is same that is mentioned for problem 1. The output of the code is mentioned below. I have used eigen built-in function in linear algebra package of julia.

Eigenvectors:

$$\begin{bmatrix} -0.607619 & 0.120058 & 0.582038 & 0.526894 \\ -0.699845 & -0.434523 & -0.559788 & -0.0896841 \\ 0.317102 & -0.866057 & 0.225527 & 0.313894 \\ -0.20115 & -0.216153 & 0.544983 & -0.784736 \end{bmatrix}$$

Matrix U:

$$\begin{bmatrix} -0.367465 & -0.25479 & 0.00715357 & -0.894427 \\ -0.73493 & -0.509581 & 0.0143071 & 0.447214 \\ 0.289206 & -0.440955 & -0.849658 & 1.26541e-8 \\ 0.491128 & -0.693518 & 0.527092 & -1.26541e-7 \end{bmatrix}$$

Matrix W:

$$\begin{bmatrix} 9.56805 & 0.0 & 0.0 & 0.0 \\ 0.0 & 5.02219 & 0.0 & 0.0 \\ 0.0 & 0.0 & 4.15091 & 0.0 \\ 0.0 & 0.0 & 0.0 & 9.05501e-8 \end{bmatrix}$$

Singular Values:

$$[9.568049054141438, 5.02219134434253, 4.150907274121396, 9.055012752297071e-8]$$

Matrix V^T:

$$\begin{bmatrix} -0.607619 & -0.699845 & 0.317102 & -0.20115 \\ 0.120058 & -0.434523 & -0.866057 & -0.216153 \\ 0.582038 & -0.559788 & 0.225527 & 0.544983 \\ 0.526894 & -0.0896841 & 0.313894 & -0.784736 \end{bmatrix}$$

Reconstructed Matrix A:

$$\begin{bmatrix} 2.0 & 3.0 & -2.0496e-8 & 1.0 \\ 4.0 & 6.0 & 1.0248e-8 & 2.0 \\ -4.0 & 1.0 & 2.0 & -2.0 \\ -2.0 & -3.0 & 5.0 & 1.0 \end{bmatrix}$$

Reconstruction Error:

$$7.300290597422665e-8$$

Inverse of Matrix W:

$$\begin{bmatrix} 0.104515 & 0 & 0 & 0 \\ 0 & 0.199116 & 0 & 0 \\ 0 & 0 & 0.240911 & 0 \\ 0 & 0 & 0 & 0.0 \end{bmatrix}$$

Residual $(Ax_p - b)$:

$$3.512737169181137e-9$$

Particular Solution (x_p):

$$\begin{bmatrix} -0.24833479718964058 \\ 0.4465250726642498 \\ 0.9371622488170107 \\ 0.15709437952841912 \end{bmatrix}$$

Null Basis:

$$\begin{bmatrix} \text{Matrix of size 4x0} \end{bmatrix}$$

General Solution:

$$\begin{bmatrix} -0.24833479718964058 \\ 0.4465250726642498 \\ 0.9371622488170107 \\ 0.15709437952841912 \end{bmatrix}$$

I have confirmed the solution satisfies the set of equations and also verified the particular solution using built in svd function in linear algebra package in julia.

Computation Time of svd function:

3.190802 seconds (5.00 M allocations: 336.226 MiB, 5.26% gc time, 99.91% compilation time)

Implications: SVD is very useful in various aspect of chemistry, physics, mathematics, computer science like Data Compression, dimensionality Reduction, eigenface Recognition ,etc problems.

Time : This code and report took nearly 6 hours i.e more than usual time as I had to figure out the eigen function gives same eigenvectors but sometimes with different sign with the in-built svd in linear algebra. ALthogh, both custom svd and in-built svd gives same solution.