Large-Scale Matrix Inversion using Singular Value Decomposition

By Vaishnavi C K

Singular Value Decomposition(SVD)

- SVD states that any m by n matrix A can be factored into

$$A=UDV^T$$

where

U=the eigen vectors of matrix AA^T

V= the eigen vectors of matrix A^TA

D=is the diagonal matrix consisting of square roots all eigen values of AA^T and A^TA

Inverse using SVD

- Therefore from SVD
- $A^{-1} = VD^{-1}U^{T}$

Where D^{-1} is the matrix containing reciprocal of all elements of D because D is diagonal matrix

Method

- Consider a matrix A
- First we compute AA^T and A^TA
- Then we find the eigen values and eigen vectors of the above matrices and name the eigen vectors as U and V respectively. Also we find that the eigen values are same for both the matrices. Thus we name the matrix as d.
- Now we take the square roots of all the values of d and name the resulting matrix as D. We find D^{-1} by reciprocating the diagonal values of D.
- Now we multiply $U D^{-1} V^T$ to find the inverse of A.

Code

```
clc;clear all;close all;
n=5;
A=rand(n)*100
 y=A*A.';
x=A.'*A;
[U, D]=eig(y);%U stores eigen vector of Y
[V, D]=eig(x);%U stores eigen vector of X
%D stores eigen value of both X and Y
for i=1:n
    d(i,i)=1/(sqrt(D(i,i)));
end
disp("U=");
disp(U)
disp("V=");
disp(V)
disp("d=");
disp(d)
inverse=V*d*U.'
error=inv(A)-inverse
```

Results

```
A =

   23.7599   96.5005   16.6012    6.8978   97.0246
   81.7571   76.5285   32.5998   16.6785   99.8427
   40.5829   57.4534   29.6436   94.7438   98.7455
   46.6312   91.5925   55.8298   81.1088   15.0087
   95.1536   49.5432    6.7477   71.0456   95.8479


y =

   1.0e+04 *

    1.9614    1.9671    1.7235    1.2889    1.6943
    1.9671    2.3850    2.0120    1.5493    2.2546
    1.7235    2.0120    2.4554    1.7976    2.3104
    1.2889    1.5493    1.7976    2.0485    1.6553
    1.6943    2.2546    2.3104    1.6553    2.5789


x =

   1.0e+04 *

    2.0124    1.9866    0.7508    1.5915    2.4296
    1.9866    2.9314    1.1248    1.8334    2.8800
    0.7508    1.1248    0.5380    0.8474    0.9277
    1.5915    1.8334    0.8474    2.0928    1.9717
    2.4296    2.8800    0.9277    1.9717    3.8545
```

Results

U=

-0.4217	0.3484	-0.5549	0.4832	0.3992
0.6077	-0.4915	-0.0836	0.3981	0.4729
0.3830	0.6972	0.2890	-0.2324	0.4792
-0.0907	-0.2662	-0.4704	-0.7437	0.3828
-0.5458	-0.2829	0.6167	0.0304	0.4907

V=

-0.0539	-0.8206	0.3773	0.0300	0.4247
-0.4231	-0.0448	-0.7404	-0.0307	0.5195
0.8652	-0.1212	-0.3375	-0.2907	0.1959
-0.1472	0.3547	0.3688	-0.7502	0.3920
0.2190	0.4290	0.2440	0.5923	0.5980

d=

0.0561	0	0	0	0
0	0.0191	0	0	0
0	0	0.0133	0	0
0	0	0	0.0107	0
0	0	0	0	0.0032

Results

```
inverse =

   -0.0063    0.0062   -0.0101    0.0024    0.0099
    0.0157   -0.0125   -0.0117    0.0079    0.0080
   -0.0201    0.0301    0.0167    0.0009   -0.0284
   -0.0002   -0.0114    0.0054    0.0031    0.0060
   -0.0003    0.0066    0.0108   -0.0088   -0.0059


error =

   1.0e-16 *

    0.1301   -0.3036   -0.0347   -0.0217    0.2082
    0.2429   -0.7980    0.0173   -0.2082    0.6939
   -0.1388    0.7980   -0.2429    0.5800   -0.8674
   -0.1163    0.1214    0.2168   -0.2602   -0.0607
   -0.1155    0.5031   -0.0867    0.1214   -0.4250
```

Power Method

- This method can be used to find the most dominant eigen value and its corresponding eigen vector.
- It is an iterative method.
- In this method we have to assume the eigen vector to be a random value and multiply with the given matrix. Now we take the product and multiply with the matrix.
- We continue multiplying the each product to the matrix until the difference between the product obtained and the previous product that was multiplied is very low.
- This product is the dominant eigen vector. And the normalizing factor we get is the dominant eigen vector.

Example

For a matrix A= $A = \begin{bmatrix} 3 & 7 \\ 4 & 5 \end{bmatrix}$

Using power method we get

$$Ay^{(1)} = x^{(0)} \rightarrow Ay^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0.1538 \\ 0.0769 \end{bmatrix} = 0.1538 \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

$$\rightarrow \alpha^{(1)} = 0.1538$$

$$Ay^{(2)} = x^{(1)} \rightarrow Ay^{(2)} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} y_1^{(2)} \\ y_2^{(2)} \end{bmatrix} = \begin{bmatrix} -0.11538 \\ 0.192307 \end{bmatrix} = 0.192307 \begin{bmatrix} -0.6 \\ 1 \end{bmatrix}$$

$$\rightarrow \alpha^{(2)} = -0.6538$$

$$Ay^{(3)} = x^{(3)} \rightarrow Ay^{(3)} = \begin{bmatrix} -0.6 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} y_1^{(3)} \\ y_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0.7692 \\ -0.4153 \end{bmatrix} = 0.7692 \begin{bmatrix} 1 \\ -0.54 \end{bmatrix}$$

$$\rightarrow \alpha^{(3)} = -0.733$$

$$Ay^{(4)} = x^{(3)} \rightarrow Ay^{(4)} = \begin{bmatrix} 1 \\ -0.54 \end{bmatrix} \rightarrow \begin{bmatrix} y_1^{(4)} \\ y_2^{(4)} \end{bmatrix} = \begin{bmatrix} -0.7292 \\ 0.4554 \end{bmatrix} = -0.7292 \begin{bmatrix} 1 \\ 0.6245 \end{bmatrix}$$

$$\rightarrow \boldsymbol{\alpha_1^{(4)} = -0.729 -> \lambda_2 = -1.371, \ldots}$$

Code

```
clc;
close all;
A=[1 3 -1;3 2 4;-1 4 10]
u= [1;1;1]; % The initial choice of eigenvector.
n=length(u); % Size of initial eigenvector.
v=zeros(n,1);
eps=0.001; %error of tolerance
err=10;m1=1;m2=1;
while err>eps  %Calculating the greatest eigenvalue and the corresponding eigenvector.
   v=A*u;
   m2=max(abs(v));
   u=v/m2;
   err=abs(m1-m2);
   m1=m2;
end
fprintf('\n The greatest eigenvalue is %5.5f\n',m1);
disp('The corresponding eigenvector is:');
fprintf('\n %5.5f\n',u);
```

Result

```
A =

     1     3    -1
     3     2     4
    -1     4    10


 The greatest eigenvalue is 11.66225
The corresponding eigenvector is:

 0.02490

 0.42174

 1.00000
```

Advantages of SVD

- The number of iterations is comparatively less compared to other conventional methods
- The inverse can be found out for any size of matrix even singular matrices.

Disadvantages using SVD

The calculation of eigen values and vectors were inefficient.

Finding the algorithm to efficiently calculate the eigen functions was difficult.