

The Singular Value Decomposition (SVD)

Both diagonalization and QR factorization have wide application in applied settings in conditions when numerical error or computational cost is an issue. Another decomposition of a matrix, the singular value decomposition, is also useful and informative.

Let A be a $n \times m$ matrix with then there exist matrices

U (orthogonal $n \times n$),

V (orthogonal $m \times m$), and

$\Sigma = \begin{bmatrix} D_{m \times m} \\ 0_{(n-m) \times m} \end{bmatrix}$ if $n \geq m$ or $\Sigma = \begin{bmatrix} D_{m \times m} & 0_{n \times (m-n)} \end{bmatrix}$ if $m > n$ note Σ is $n \times m$

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

Where $D_{ij} = \sigma_i \delta_{ij}$ and $\sigma_i = \sqrt{\lambda_i}$, which are called singular values of A and are the square roots of the positive eigen values of $A^T A$ or AA^T , which are the same. Usually ordered from greatest to least down the diagonal.

The columns of U are the orthogonal and normalized eigen vectors of AA^T which are called the left singular vectors of A ordered to be paired up with σ_i

The columns of V are the orthogonal and normalized eigen vectors of $A^T A$ which are called the right singular vectors of A ordered to be paired up with σ_i

Example: Find the singular value decomposition of $A = \begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix}$

The Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD)

```
octave:71> A=[1 2; 2 0; 0 2]
A =
```

$$\begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix}$$

```
octave:72> [U1, DU1]=eig(A*A')
U1 =
```

$$\begin{bmatrix} 2/3 & 0 & 963/1292 \\ -1/3 & -305/341 & 963/3230 \\ -2/3 & 1292/2889 & 963/1615 \end{bmatrix}$$

```
DU1 =
```

Diagonal Matrix

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

```
octave:73> Sigma=[3 0; 0 2; 0 0]
Sigma =
```

$$\begin{bmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix}$$

```
octave:74> U=[U1(:,3) U1(:,2) U1(:,1)]
U =
```

$$\begin{bmatrix} 963/1292 & 0 & 2/3 \\ 963/3230 & -305/341 & -1/3 \\ 963/1615 & 1292/2889 & -2/3 \end{bmatrix}$$

```
octave:75> [V1, DV1]=eig(A'*A)
V1 =
```

$$\begin{bmatrix} -305/341 & 1292/2889 \\ 1292/2889 & 5473/6119 \end{bmatrix}$$

```
DV1 =
```

Diagonal Matrix

$$\begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$$

```
octave:76> V=[V1(:,2) V1(:,1)]
V =
```

$$\begin{bmatrix} 1292/2889 & -305/341 \\ 5473/6119 & 1292/2889 \end{bmatrix}$$

```
octave:77> U*Sigma*V'
ans =
```

$$\begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The Singular Value Decomposition (SVD)

Applications of the SVD



Image Compression: We can think of the image above as a matrix where we assign a gray scale intensity to each pixel position. So we have a matrix of numbers between 0 and 255.

If you recall, in the matrix Σ is ordered with the largest singular values first and going down from there. Our SVD product can be written in the form

$$A = U\Sigma V^T = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_n \vec{u}_n \vec{v}_n^T$$

$\vec{u}_i$ is $n \times 1$ $\vec{v}_i^T$ is $1 \times m$ so $\vec{u}_i \vec{v}_i^T$ is $n \times m$ the same as A

From this we might note that if the smallest singular values are very small, they don't contribute much information to A . So what would happen to the image if we throw away some of terms being scaled by small singular values?

$$\text{So we might say: } A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_k \vec{u}_k \vec{v}_k^T$$

Where $k < n$ keeping in mind that $\sigma_1 > \sigma_2 > \sigma_3 > \cdots > \sigma_n$

In matlab matrix multiplication notation keeping the first 10 terms would read

$$U(:,1:10)S(1:10,1:10)V(:,1:10)'$$

The Singular Value Decomposition (SVD)



Keeping only 10 singular values

$U(:,1:10)S(1:10,1:10)V(:,1:10)'$



Keeping only 50 singular values

$U(:,1:50)S(1:50,1:50)V(:,1:50)'$



Keeping only 100 singular values

$U(:,1:100)S(1:100,1:100)V(:,1:100)'$



Original 562X562

The Singular Value Decomposition (SVD)

The condition number of a matrix

The condition number of a matrix A is $\kappa(A) = \frac{\sigma_{\max}}{\sigma_{\min}}$ This number gives a sense of how terrible an inversion of a matrix might be!

```
>> B=randi([-10 10], 5, 3)
```

B =

-2	-4	-8
1	-1	-3
-2	-10	-6
3	10	0
3	-7	-3

```
>> A=[B B(:,2)-0.001, B(:,1)-.01]
```

A =

-2	-4	-8	-4001/1000	-201/100
1	-1	-3	-1001/1000	99/100
-2	-10	-6	-10001/1000	-201/100
3	10	0	9999/1000	299/100
3	-7	-3	-7001/1000	299/100

```
>> [U,S,V]=svd(A)
```

U =

-431/1274	-759/1090	385/1053	569/1102	302/15201
-427/4964	-421/1285	-229/1069	-466/1487	-588/683
-1139/1791	-231/1943	239/1902	-268/395	306/943
1171/2076	-3543/5656	-1316/4877	-339/1138	604/1689
-941/2386	-356/10045	-2995/3501	547/1864	305/1919

S =

8644/353	0	0	0	0
0	4762/609	0	0	0
0	0	2905/442	0	0
0	0	0	59/17298	0
0	0	0	0	*

V =

295/3046	-243/2783	-997/1433	-272/387	-221/3141
563/851	-415/1893	269/2252	-538/7655	489/695
1107/3404	2049/2173	-511/7148	-30/19169	*
698/1055	-966/4411	295/2467	188/2669	-489/695
265/2726	-1028/12093	-2441/3515	2785/3954	221/3141

```
>> svd(A)
```

ans =

8644/353
4762/609
2905/442
59/17298
1/5614388000771653

This condition number is huge!
We would expect any inversion to be horrible!

```
>> S(1,1)/S(5,5)
```

ans =

137480924864676784

The Singular Value Decomposition (SVD)

```
>> x=[1;2;3;4;5]

x =

     1
     2
     3
     4
     5

>> b=A*x

b =

-60.0540
 -9.0540
-90.0540
 77.9460
-33.0540

>> inv(A)*b
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 3.663224e-18.

ans =

 1.0396
 3.5995
 3.0000
12.4005
 3.4275
```

Why do we have this problem? Let's think about inverting the SVD version of A....

$$A\vec{x} = \vec{b}$$

$$(USV^T)\vec{x} = \vec{b}$$