Table of Contents

DIP Assign4 - Q.1

```
% A: Input Matrix
% Areconstructed: reconstructed matrix using mySVD routine
% Snew: singular values using mySVD routine
% SOrig: singular values using MATLAB's SVD function
% U: Left singular vectors using mySVD routine
% UOrig:  Left singular vectors using MATLAB's SVD function
% V: Right singular vectors using mySVD routine
% VOrig: Right singular vectors using MATLAB's SVD function
```

Case 1: m = n

```
A = rand(5,5);
[Areconstructed,U,Snew,V,UOrig,SOrig,VOrig] = mySVD(A,5,5);
A
Areconstructed
Snew
SOrig
U
UOrig
V
VOrig
```

```
A =

    0.8530    0.6403    0.1057    0.0521    0.8604
    0.8739    0.4170    0.1420    0.9312    0.9344
    0.2703    0.2060    0.1665    0.7287    0.9844
    0.2085    0.9479    0.6210    0.7378    0.8589
    0.5650    0.0821    0.5737    0.0634    0.7856


Areconstructed =

    0.8530    0.6403    0.1057    0.0521    0.8604
    0.8739    0.4170    0.1420    0.9312    0.9344
    0.2703    0.2060    0.1665    0.7287    0.9844
    0.2085    0.9479    0.6210    0.7378    0.8589
    0.5650    0.0821    0.5737    0.0634    0.7856
```

```
Snew =

    2.9167         0         0         0         0
         0    0.8479         0         0         0
         0         0    0.7010         0         0
         0         0         0    0.5628         0
         0         0         0         0    0.2913


SOrig =

    2.9167         0         0         0         0
         0    0.8479         0         0         0
         0         0    0.7010         0         0
         0         0         0    0.5628         0
         0         0         0         0    0.2913


U =

    0.4197    0.6182   -0.2540    0.4854   -0.3762
    0.5401    0.0072    0.5891    0.2541    0.5446
    0.4103   -0.2693    0.4108   -0.3334   -0.6923
    0.5053   -0.5910   -0.6097    0.0922    0.1231
    0.3293    0.4426   -0.2189   -0.7617    0.2599


UOrig =

   -0.4197    0.6182    0.2540   -0.4854   -0.3762
   -0.5401    0.0072   -0.5891   -0.2541    0.5446
   -0.4103   -0.2693   -0.4108    0.3334   -0.6923
   -0.5053   -0.5910    0.6097   -0.0922    0.1231
   -0.3293    0.4426    0.2189    0.7617    0.2599


V =

    0.4225    0.6932    0.2260    0.2396    0.4822
    0.3718   -0.2129   -0.6109    0.6628   -0.0628
    0.2373   -0.1079   -0.5406   -0.6180    0.5078
    0.4174   -0.6668    0.5292    0.0689    0.3105
    0.6728    0.1341    0.0580   -0.3415   -0.6398


VOrig =

   -0.4225    0.6932   -0.2260   -0.2396    0.4822
   -0.3718   -0.2129    0.6109   -0.6628   -0.0628
   -0.2373   -0.1079    0.5406    0.6180    0.5078
   -0.4174   -0.6668   -0.5292   -0.0689    0.3105
   -0.6728    0.1341   -0.0580    0.3415   -0.6398
```

Case 2: m > n

```
A = rand(6,4);
[Areconstructed,U,Snew,V,UOrig,SOrig,VOrig] = mySVD(A,6,4);
A
Areconstructed
Snew
SOrig
U
UOrig
V
VOrig
```

```
A =

    0.5134    0.3013    0.8422    0.1771
    0.1776    0.2955    0.5590    0.6628
    0.3986    0.3329    0.8541    0.3308
    0.1339    0.4671    0.3479    0.8985
    0.0309    0.6482    0.4460    0.1182
    0.9391    0.0252    0.0542    0.9884


Areconstructed =

    0.5134    0.3013    0.8422    0.1771
    0.1776    0.2955    0.5590    0.6628
    0.3986    0.3329    0.8541    0.3308
    0.1339    0.4671    0.3479    0.8985
    0.0309    0.6482    0.4460    0.1182
    0.9391    0.0252    0.0542    0.9884


Snew =

    2.2218         0         0         0
         0    1.0981         0         0
         0         0    0.7004         0
         0         0         0    0.3200
         0         0         0         0
         0         0         0         0


SOrig =

    2.2218         0         0         0
         0    1.0981         0         0
         0         0    0.7004         0
         0         0         0    0.3200
         0         0         0         0
         0         0         0         0
```

```
U =

    0.4068    0.3497   -0.5163   -0.0176   -0.5957    0.3009
    0.4050    0.0488    0.2544    0.5076   -0.2656   -0.6638
    0.4350    0.3291   -0.2804    0.2281    0.7542    0.0548
    0.4362   -0.1171    0.6503    0.1050   -0.0342    0.6008
    0.2507    0.4268    0.3036   -0.7589    0.0330   -0.2929
    0.4777   -0.7558   -0.2739   -0.3208    0.0596   -0.1382


UOrig =

   -0.4068   -0.3497   -0.5163   -0.0176    0.1040   -0.6592
   -0.4050   -0.0488    0.2544    0.5076    0.6939    0.1724
   -0.4350   -0.3291   -0.2804    0.2281   -0.4853    0.5799
   -0.4362    0.1171    0.6503    0.1050   -0.4675   -0.3789
   -0.2507   -0.4268    0.3036   -0.7589    0.2184    0.1979
   -0.4777    0.7558   -0.2739   -0.3208    0.0774    0.1291


V =

    0.4361   -0.3579   -0.7030   -0.4331
    0.3445    0.3936    0.4567   -0.7196
    0.5536    0.6479   -0.2646    0.4514
    0.6202   -0.5452    0.4767    0.3013


VOrig =

   -0.4361    0.3579   -0.7030   -0.4331
   -0.3445   -0.3936    0.4567   -0.7196
   -0.5536   -0.6479   -0.2646    0.4514
   -0.6202    0.5452    0.4767    0.3013
```

Case 13: m < n

```
A = rand(4,6);
[Areconstructed,U,Snew,V,UOrig,SOrig,VOrig] = mySVD(A,4,6);
A
Areconstructed
Snew
SOrig
U
UOrig
V
VOrig
```

```
A =
```

```
    0.5400    0.4145    0.1002    0.5219    0.9052    0.1040
    0.7069    0.4648    0.1781    0.3358    0.6754    0.7455
    0.9995    0.7640    0.3596    0.1757    0.4685    0.7363
    0.2878    0.8182    0.0567    0.2089    0.9121    0.5619


Areconstructed =

    0.5400    0.4145    0.1002    0.5219    0.9052    0.1040
    0.7069    0.4648    0.1781    0.3358    0.6754    0.7455
    0.9995    0.7640    0.3596    0.1757    0.4685    0.7363
    0.2878    0.8182    0.0567    0.2089    0.9121    0.5619


Snew =

    2.6529         0         0         0         0         0
         0    0.7409         0         0         0         0
         0         0    0.4930         0         0         0
         0         0         0    0.2636         0         0


SOrig =

    2.6529         0         0         0         0         0
         0    0.7409         0         0         0         0
         0         0    0.4930         0         0         0
         0         0         0    0.2636         0         0


U =

    0.4238    0.6258    0.6207    0.2085
    0.5064   -0.1693    0.1066   -0.8388
    0.5652   -0.6558    0.1115    0.4878
    0.4944    0.3867   -0.7688    0.1228


UOrig =

   -0.4238   -0.6258   -0.6207    0.2085
   -0.5064    0.1693   -0.1066   -0.8388
   -0.5652    0.6558   -0.1115    0.4878
   -0.4944   -0.3867    0.7688    0.1228


V =

    0.4878   -0.4399    0.6099    0.1612    0.1315   -0.3916
    0.4702   -0.0052   -0.4808    0.6435   -0.3653   -0.0149
    0.1372   -0.2448    0.1576    0.2044    0.2972    0.8754
    0.2239    0.3177    0.4436   -0.2336   -0.7229    0.2739
    0.5433    0.6717   -0.0308   -0.1416    0.4818   -0.0223
    0.4205   -0.4410   -0.4176   -0.6660   -0.0787    0.0682
```

```
VOrig =

   -0.4878    0.4399   -0.6099    0.1612   -0.3006    0.2833
   -0.4702    0.0052    0.4808    0.6435   -0.1663   -0.3255
   -0.1372    0.2448   -0.1576    0.2044    0.9194   -0.0964
   -0.2239   -0.3177   -0.4436   -0.2336   -0.0538   -0.7712
   -0.5433   -0.6717    0.0308   -0.1416    0.1814    0.4469
   -0.4205    0.4410    0.4176   -0.6660    0.0290   -0.1001
```

Comments

Concerns with finding U, V and S directly as eigen values and eigen vectors of AA' and A'A: 1. The order of columns of U, V and S is exactly opposite of that of original U, V and S (but that is not a major concern as far as the three of them have consistent ordering). 2. If x is an eigen vector of A, so is -x; and so U and V might have one or more columns as negative of the columns of original U and V. Even here, if all columns of both U and V are negated it is not a concern.

Way to deal with these concerns: For instance, A = USV' ==> AV = US ==> U'A = SV' We will use one of the two expressions depending on whether m >/</= n. S is a diagonal matrix: So, U will simply be columns of AV divided by corresponding diagonal elements of S, as far as they are non-zero. OR V will simply be rows of U'A divided by corresponding diagonal elements of S, as far as they are non-zero.

Published with MATLAB® R2014a