Project Two Template

MAT-350: Applied Linear Algebra

Matthew Muller

12/8/21

Problem 1

Use the svd() function in MATLAB to compute A_1, the **rank-1 approximation of** A**.** Clearly state what A_1 is, rounded to 4 decimal places. Also, **compute** the root-mean square error (RMSE) between A and A_1.

Solution:

```
%code
%Create matrix A
A = [1 2 2; 3 4 5; 6 7 8]
```

```
A = 3×3
     1     2     2
     3     4     5
     6     7     8
```

```
%Perform the svd function on A and put results into U, S, and V
[U, S, V] = svd(A)
```

```
U = 3×3
   -0.2055   -0.6658   -0.7172
   -0.4900   -0.5644    0.6643
   -0.8471    0.4880   -0.2103
S = 3×3
   14.4042         0         0
         0    0.6450         0
         0         0    0.3229
V = 3×3
   -0.4692    0.8820    0.0433
   -0.5763   -0.2687   -0.7718
   -0.6691   -0.3871    0.6344
```

```
%Calculate Rank 1 approximation of A
A1 = U(:,1:1) * S(1:1, 1:1) * V(:,1:1)'
```

```
A1 = 3×3
    1.3889    1.7059    1.9807
    3.3118    4.0678    4.7230
    5.7253    7.0322    8.1649
```

```
%Verify rank = 1
rank(A1)
```

```
ans = 1
```

```
%Calculate root-mean square error (RMSE) between A and A1
RMSE1 = norm(A-A1,'fro')/(3*3)
```

```
RMSE1 = 0.0801
```

Problem 2

Use the svd() function in MATLAB to compute A_2, the **rank-2 approximation of** A. Clearly state what A_2 is, rounded to 4 decimal places. Also, **compute** the root-mean square error (RMSE) between A and A_2. Which approximation is better, A_1 or A_2? Explain.

Solution:

```
%code
%Calculate Rank 2 approximation of A
A2 = U(:,1:2) * S(1:2, 1:2) * V(:,1:2)'
```

```
A2 = 3×3
    1.0100    1.8213    2.1469
    2.9907    4.1656    4.8639
    6.0029    6.9476    8.0431
```

```
%Verify rank = 2
rank(A2)
```

```
ans = 2
```

```
%Calculate root-mean square error between A and A2
RMSE2 = norm(A-A2,'fro')/(3*3)
```

```
RMSE2 = 0.0359
```

Explain:

A2 is the better approximation because it has a lower RMSE. As shown in the project 2 SVD document, as we go into higher rank approximations, the lower the RMSE becomes.

Problem 3

For the 3×3 matrix A, the singular value decomposition is $A = USV'$ where $U = [\mathbf{u}_1 \mathbf{u}_2 \mathbf{u}_3]$. Use MATLAB to **compute** the dot product $d_1 = dot(\mathbf{u}_1, \mathbf{u}_2)$.

Also, use MATLAB to **compute** the cross product $\mathbf{c} = cross(\mathbf{u}_1, \mathbf{u}_2)$ and dot product $d_2 = dot(\mathbf{c}, \mathbf{u}_3)$. Clearly state the values for each of these computations. Do these values make sense? **Explain**.

Solution:

```
%code
%Separate U into three different vectors in order to calculate cross and
%dot products
U1 = [-0.2055; -0.4900; -0.8471]
```

```
U1 = 3×1
   -0.2055
   -0.4900
```

```
   -0.8471
```

```
U2 = [-0.6658; -0.5644; 0.4880]
```

```
U2 = 3×1
   -0.6658
   -0.5644
    0.4880
```

```
U3 = [-0.7172; 0.6643; -0.2103]
```

```
U3 = 3×1
   -0.7172
    0.6643
   -0.2103
```

```
%Calculate dot product d1
d1 = dot(U1,U2)
```

```
d1 = -6.9000e-06
```

```
%Calculate cross product
c = cross(U1,U2)
```

```
c = 3×1
   -0.7172
    0.6643
   -0.2103
```

```
%Calculate dot product d2
d2 = dot(c,U3)
```

```
d2 = 0.9999
```

Explain:

The cross product of U1 and U2, c, makes sense as it equals U3. This shows that U1 and U2 are perpendicular vectors. The dot product d2 also makes sense being 0.9999 (essentially 1) since c and U3 are the exact same.

Problem 4

Using the matrix $U = [\mathbf{u}_1 \mathbf{u}_2 \mathbf{u}_3]$, d**etermine whether or not the columns of** U **span** $\mathbb{R}^3$. **Explain your approach.**

Solution:

```
%code
%Put the column U1, U2, and U3 vectors back into U
U = [U1 U2 U3]
```

```
U = 3×3
   -0.2055   -0.6658   -0.7172
   -0.4900   -0.5644    0.6643
   -0.8471    0.4880   -0.2103
```

```
%Row reduce matrix U
reducedU = rref(U)
```

```
reducedU = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
%Check the rank of reducedU matrix
rank(reducedU)
```

```
ans = 3
```

Explain:

Since the reduced matrix reducedU has 3 pivot columns and therefore a rank of 3, the columns of U do in fact span R3.

Problem 5

Use the MATLAB imshow() function to load and display the image A stored in the image.mat file, available in the Project Two Supported Materials area in Brightspace. For the loaded image, **derive the value of** k that will result in a compression ratio of $CR \approx 2$. For this value of k, **construct the rank-*k* approximation of the image**.

Solution:

```
%code
%Load the image
load image.mat;
%Display the image
imshow(A)
```

```
%Perform svd function on A, putting results into U, S and V
[U, S, V]= svd(double(A))
```

U = 3072×3072

-0.0220	0.0337	-0.0276	0.0071	-0.0003	0.0114	-0.0108	0.0043 ···
-0.0220	0.0335	-0.0273	0.0066	-0.0002	0.0106	-0.0112	0.0037
-0.0220	0.0335	-0.0271	0.0062	-0.0003	0.0100	-0.0113	0.0029
-0.0220	0.0333	-0.0271	0.0057	-0.0003	0.0094	-0.0110	0.0023
-0.0219	0.0331	-0.0273	0.0053	-0.0003	0.0083	-0.0109	0.0020
-0.0219	0.0329	-0.0274	0.0049	-0.0007	0.0066	-0.0107	0.0017
-0.0219	0.0325	-0.0274	0.0041	-0.0012	0.0048	-0.0106	0.0012
-0.0218	0.0322	-0.0277	0.0037	-0.0012	0.0027	-0.0104	0.0008
-0.0218	0.0321	-0.0281	0.0028	-0.0020	0.0008	-0.0097	0.0003
-0.0218	0.0319	-0.0281	0.0020	-0.0025	-0.0009	-0.0093	-0.0001
⋮							

S = 3072×4608

10^5 ×

5.7986	0	0	0	0	0	0	0 ···
0	0.6755	0	0	0	0	0	0
0	0	0.3657	0	0	0	0	0
0	0	0	0.3129	0	0	0	0
0	0	0	0	0.2842	0	0	0
0	0	0	0	0	0.2423	0	0
0	0	0	0	0	0	0.2325	0
0	0	0	0	0	0	0	0.2217
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
⋮							

V = 4608×4608

-0.0159	-0.0085	-0.0079	-0.0083	0.0064	-0.0076	0.0061	0.0098 ···

```
   -0.0159   -0.0087   -0.0081   -0.0084    0.0064   -0.0082    0.0064    0.0100
   -0.0159   -0.0088   -0.0079   -0.0085    0.0063   -0.0078    0.0067    0.0104
   -0.0160   -0.0090   -0.0081   -0.0087    0.0063   -0.0077    0.0069    0.0106
   -0.0160   -0.0091   -0.0079   -0.0092    0.0058   -0.0077    0.0071    0.0104
   -0.0160   -0.0092   -0.0082   -0.0091    0.0055   -0.0076    0.0076    0.0104
   -0.0160   -0.0093   -0.0081   -0.0094    0.0054   -0.0078    0.0076    0.0106
   -0.0160   -0.0094   -0.0083   -0.0094    0.0053   -0.0076    0.0080    0.0107
   -0.0160   -0.0095   -0.0084   -0.0098    0.0053   -0.0075    0.0080    0.0108
   -0.0160   -0.0097   -0.0085   -0.0099    0.0052   -0.0074    0.0081    0.0112
     ⋮
```

```
%Find value of K that provides a CR of approximate 2
%CR = (M*N)/K(M+N+1)
%2 = (3072*4608)/K(3072+4608+1)
CR = (3072*4608) / (921 * (3072+4608+1))
```

```
CR = 2.0010
```

```
%Calculate rank-921 approximation of A
A921 = U(:, 1:921)*S(1:921, 1:921)*V(:,1:921)'
```

```
A921 = 3072×4608
  189.0646  191.8936  188.8820  187.8085  190.9213  193.6283  196.9239  193.2751 ···
  188.8330  192.1524  189.7149  190.0349  191.8535  193.0206  196.4297  194.3539
  189.4446  192.6200  190.0603  190.5890  191.3810  192.0473  197.1031  196.4040
  191.2359  192.9759  190.2975  191.5527  190.4033  189.8999  196.5573  197.9993
  191.6491  193.2942  190.5409  193.6870  191.5410  189.6225  195.2290  197.1438
  190.4965  192.6234  190.1108  194.4575  193.5966  190.9228  195.0517  194.8664
  188.1900  191.7679  191.4803  193.0678  193.5434  192.0132  195.9747  195.3500
  188.3936  192.2560  192.5138  192.7716  192.9265  193.7176  196.6674  196.9272
  191.2526  192.7502  192.4498  192.8418  193.6039  194.3199  198.5062  199.0176
  192.6733  194.3953  194.4557  193.9848  194.1313  194.8077  197.6567  198.6137
     ⋮
```

```
%Verify Rank = 921
rank(A921)
```

```
ans = 921
```

```
%Convert the image back into its original format, uint8
A921 = uint8(round(A921))
```

```
A921 = 3072×4608 uint8 matrix
   189   192   189   188   191   194   197   193   196   195   190   193   192 ···
   189   192   190   190   192   193   196   194   198   197   191   193   190
   189   193   190   191   191   192   197   196   198   198   192   192   188
   191   193   190   192   190   190   197   198   197   197   193   193   190
   192   193   191   194   192   190   195   197   195   196   193   196   193
   190   193   190   194   194   191   195   195   195   198   196   198   193
   188   192   191   193   194   192   196   195   197   198   196   199   194
   188   192   193   193   193   194   197   197   199   199   195   199   195
   191   193   192   193   194   194   199   199   199   199   197   200   193
   193   194   194   194   194   195   198   199   199   198   196   197   192
     ⋮
```

Explain:

I first loaded the image to find that the dimensions are M = 3072 and N = 4608. I then converted this matrix from a uint8 to a double so that I could use the svd function on A. After performing the svd function, I used the equation CR = MN/ 6 K(M + N + 1) to find the value of K that yields a compression ratio of 2. I found that K was 921. I then calculated the rank-921 approximation of A using U, S, and V. I then verified that the rank of this approximation was in fact 921 and converted the A921 matrix back to a uint8 so that I can display the image of the Rank-921 approximation in Problem 6.

Problem 6

Display the image and compute the root mean square error (RMSE) between the approximation and the original image. Make sure to include a copy of the approximate image in your report.

Solution:

```
%code
%Show the image of the Rank-921 approximation of A
imshow(A921)
```

```
%Calculate RSME between A and A-921
RSME921 = norm(double(A)-double(A921),'fro')/(3072 * 4608)
```

```
RSME921 = 4.0658e-04
```

Problem 7

Repeat Problems 5 and 6 for $CR \approx 10$, $CR \approx 25$, and $CR \approx 75$. **Explain** what trends you observe in the image approximation as CR increases and provide your recommendation for the best CR based on your observations. Make sure to include a copy of the approximate images in your report.

Solution:

```
%code
%Find value of K that provides a CR of around 10
%CR = (M*N)/K(M+N+1)
%10 = (3072*4608)/K(3072+4608+1)
CR = (3072*4608) / (184*(3072+4608+1))
```

```
CR = 10.0161
```

```
%Calculate rank-184 approximation of A
A184 = U(:, 1:184)*S(1:184, 1:184)*V(:,1:184)'
```

```
A184 = 3072×4608
  189.7432  191.5483  189.1171  188.5879  189.2042  190.4093  192.2473  192.2744 ···
  189.9357  191.6766  189.2152  189.0111  189.4455  190.7028  192.9499  193.0818
  190.3505  191.8218  189.2282  189.5003  189.7682  190.8722  193.6113  194.0608
  190.4580  191.6506  189.2845  190.0465  190.6301  191.5918  194.6700  195.6617
  189.7958  190.8026  188.7556  190.2446  190.8688  191.7150  195.1483  196.7306
  189.5912  190.5549  188.8867  190.6705  191.0324  191.8086  195.1119  196.9319
  189.8759  190.7995  189.5386  191.3636  191.7975  192.5087  195.5045  197.6150
  190.6872  191.5916  190.2235  192.3257  192.6240  193.2553  195.9026  198.1159
  192.4771  193.3784  192.0164  194.2350  194.6985  195.1316  197.0148  198.9931
  194.2754  195.2644  193.8108  195.8900  195.8527  196.3854  197.4867  199.1781
      ⋮
```

```
%Verify rank = 184
rank(A184)
```

```
ans = 184
```

```
%Convert the image back into its original format, uint8
A184 = uint8(round(A184))
```

```
A184 = 3072×4608 uint8 matrix
   190   192   189   189   189   190   192   192   195   195   193   195   192 ···
   190   192   189   189   189   191   193   193   195   195   193   196   193
   190   192   189   190   190   191   194   194   196   196   193   196   193
   190   192   189   190   191   192   195   196   197   197   195   197   194
   190   191   189   190   191   192   195   197   198   198   196   198   195
   190   191   189   191   191   192   195   197   198   198   196   198   194
   190   191   190   191   192   193   196   198   199   199   196   199   194
   191   192   190   192   193   193   196   198   199   199   197   199   194
   192   193   192   194   195   195   197   199   200   200   198   200   195
   194   195   194   196   196   196   197   199   200   200   198   200   194
     ⋮
```

```
%Display the image the Rank-184 approximation of A
imshow(A184)
```

```
%Calculate RSME between A and A-184
RSME184 = norm(double(A)-double(A184),'fro')/(3072 * 4608)
```

RSME184 = 9.1735e-04

```
%Find value of K that provides a CR of around 25
%CR = (M*N)/K(M+N+1)
%25 = (3072*4608)/K(3072+4608+1)
CR = (3072*4608) / (74*(3072+4608+1))
```

CR = 24.9049

```
%Calculate rank-74 approximation of A
A74 = U(:, 1:74)*S(1:74, 1:74)*V(:,1:74)'
```

A74 = 3072×4608

195.2146	194.2607	194.6176	195.0690	195.0659	195.4609	196.9847	196.5227	⋯
195.0544	194.0670	194.3006	194.8159	194.8360	195.1776	196.6900	196.1473	
194.9326	193.9499	194.1028	194.6309	194.7149	194.9625	196.5167	195.8726	
194.6416	193.6869	193.8314	194.3918	194.5278	194.7520	196.3028	195.6323	
194.2353	193.2686	193.3772	193.9606	194.2055	194.3836	195.9494	195.2664	
193.4457	192.4704	192.5364	193.0812	193.4071	193.5124	195.1114	194.4368	
192.9854	191.9917	192.0466	192.5400	192.8802	192.9862	194.5540	193.8953	
192.0525	191.1378	191.2363	191.7405	192.1727	192.2654	193.8367	193.2267	
191.1395	190.2445	190.3425	190.8276	191.2458	191.3305	192.8443	192.2891	
189.9394	189.1318	189.2626	189.7751	190.2202	190.2856	191.8472	191.2855	
⋮								

```
%Verify rank = 74
```

```
rank(A74)
```

```
ans = 74
```

```
%Convert the image back into its original format, uint8
A74 = uint8(round(A74))
```

A74 = 3072×4608 uint8 matrix

195	194	195	195	195	195	197	197	197	198	197	199	196 ⋯
195	194	194	195	195	195	197	196	197	198	197	198	196
195	194	194	195	195	195	197	196	196	197	197	198	195
195	194	194	194	195	195	196	196	196	197	196	198	195
194	193	193	194	194	194	196	195	196	197	196	197	195
193	192	193	193	193	194	195	194	195	196	195	196	193
193	192	192	193	193	193	195	194	194	195	195	196	193
192	191	191	192	192	192	194	193	194	195	194	195	192
191	190	190	191	191	191	193	192	193	194	193	194	191
190	189	189	190	190	190	192	191	192	193	192	193	190
⋮												

```
%Display the image the Rank-74 approximation of A
imshow(A74)
```

```
%Calculate RSME between A and A-74
RMSE4 = norm(double(A)-double(A74),'fro')/(3072*4608)
```

```
RMSE4 = 0.0018
```

```
%Find value of K that provides a CR of around 75
%CR = (M*N)/K(M+N+1)
```

```
%75 = (3072*4608)/K(3072+4608+1)
CR = (3072*4608) / (25*(3072+4608+1))
```

```
CR = 73.7184
```

```
%Calculate rank-25 approximation of A
A25 = U(:, 1:25)*S(1:25, 1:25)*V(:,1:25)'
```

```
A25 = 3072×4608
  183.8670  183.5431  184.1842  184.1103  183.9570  183.8832  184.7547  184.9945 ...
  182.6307  182.3624  183.0260  182.9594  182.8929  182.8136  183.7167  183.9619
  181.1605  180.9303  181.6012  181.5267  181.5471  181.4601  182.3846  182.6361
  180.2502  180.0614  180.7289  180.6456  180.7168  180.6368  181.5718  181.8339
  179.3194  179.2104  179.8921  179.8020  179.9650  179.9045  180.8685  181.1550
  178.0185  178.0003  178.6870  178.5843  178.8720  178.8329  179.8312  180.1462
  176.7755  176.8469  177.5490  177.4406  177.8662  177.8534  178.8894  179.2291
  175.4634  175.6402  176.3655  176.2486  176.8075  176.8247  177.9027  178.2709
  174.8463  175.1319  175.8624  175.7428  176.4252  176.4820  177.6023  178.0109
  174.2013  174.5574  175.2860  175.1742  175.9547  176.0315  177.1715  177.6014
     ⋮
```

```
%Verify rank = 25
rank(A25)
```

```
ans = 25
```

```
%Convert the image back into its original format, uint8
A25 = uint8(round(A25))
```

```
A25 = 3072×4608 uint8 matrix
   184   184   184   184   184   184   185   185   185   185   186   186   185 ...
   183   182   183   183   183   183   184   184   184   184   185   185   184
   181   181   182   182   182   181   182   183   183   183   183   184   183
   180   180   181   181   181   181   182   182   182   182   183   183   182
   179   179   180   180   180   180   181   181   181   182   182   182   182
   178   178   179   179   179   179   180   180   180   181   181   181   181
   177   177   178   177   178   178   179   179   180   180   180   181   180
   175   176   176   176   177   177   178   178   179   179   179   180   179
   175   175   176   176   176   176   178   178   178   179   179   179   179
   174   175   175   175   176   176   177   178   178   178   179   179   179
     ⋮
```

```
%Display the image the Rank-25 approximation of A
imshow(A25)
```

Explain:

As the CR value increases, so does the RMSE while the value of k decreases and the image becomes more blurry. For CR values of 25 and 75, the image is extremely blurry but for a CR value of 10, the image is reasonably clear. Because of this, I would recommend a CR of approximately 10 in order to maximize both image quality and space saving.