Table of Contents

Project 1 Image Classification using SVD Tylor Cooks

```
clear all, close all, clc   % all 4 files and code in the same
 directory
```

Load samples from MNIST Dataset

Load MNIST dataset

```
[train_imgs, train_labels] = readMNIST('train-images.idx3-
ubyte', 'train-labels.idx1-ubyte', 60000, 0);
[test_imgs, test_labels] = readMNIST('t10k-images.idx3-ubyte', 't10k-
labels.idx1-ubyte', 10000, 0);
```

Parameters and Reshape of Images

```
% Parameters
shape_of_image_data = size(train_imgs);
n = shape_of_image_data(1); %row
m = shape_of_image_data(2); %column

% Reshape train images into column vectors
num_images = size(train_imgs, 3);
image_size = size(train_imgs, 1) * size(train_imgs, 2);
reshaped_train_images = reshape(train_imgs, image_size, num_images);

% Reshape test images into column vectors
num_images = size(test_imgs, 3);
image_size = size(test_imgs, 1) * size(test_imgs, 2);
reshaped_test_images = reshape(test_imgs, image_size, num_images);
```

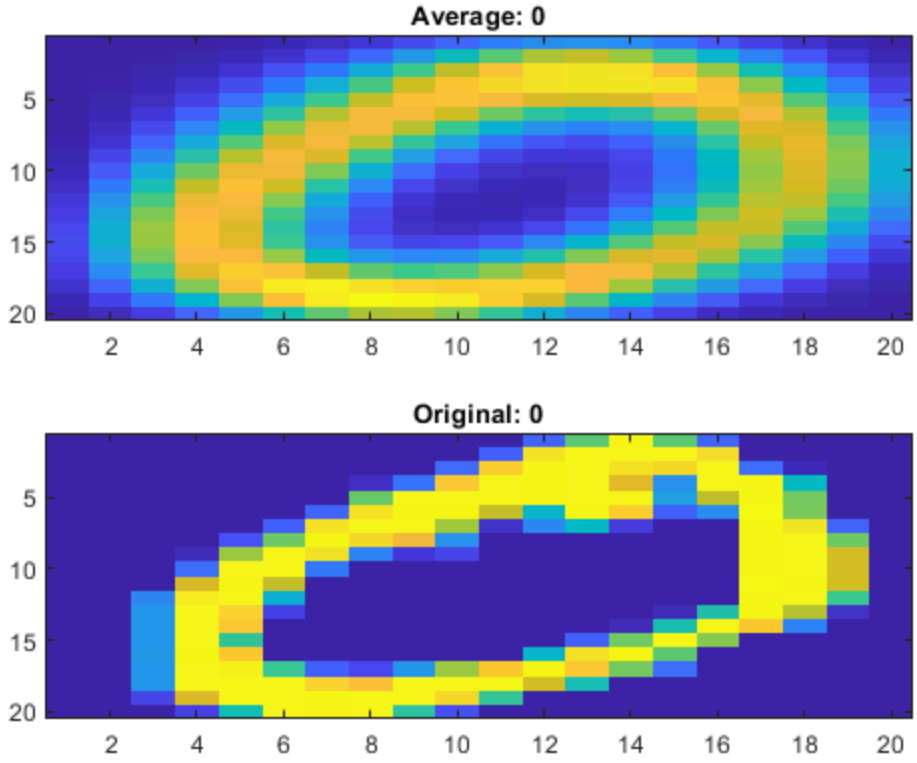
Sort data, Digits 0-9, Caluclate Avg

```
database_trainImg = cell(3,10); % database_trainImg to hold the
 original and mean of the digits
```

```
for i = 0:9
    digit_indices = find(train_labels == i);
    train_images = reshaped_train_images(:, digit_indices);
    digit_mean = mean(train_images,2);
    X_MS = train_images-digit_mean;
    database_trainImg{1,i+1} = train_images;
    database_trainImg{2,i+1} = X_MS;
    database_trainImg{3,i+1} = digit_mean;
end

database_testImg = cell(1,10);
for i = 0:9
    test_indices = find(test_labels == i);
    test_images = reshaped_test_images(:, test_indices);
    database_testImg{1,i+1} = test_images;
end
img_sort =1; % sorted data, 0-9
MS = 2; % mean subtracted
avg_img = 3; % average of the image
```

Compare the Average to an Original Image, Digit 0

```
% plot average digit 0 vs an original digit 0 from database_trainImg
figure(1)
subplot(2,1,1);
imagesc(reshape(database_trainImg{avg_img,1},[20,20]))
title('Average: 0');
subplot(2,1,2);
imagesc(reshape(database_trainImg{img_sort,1}(:,1),[20,20]))
title('Original: 0');
```



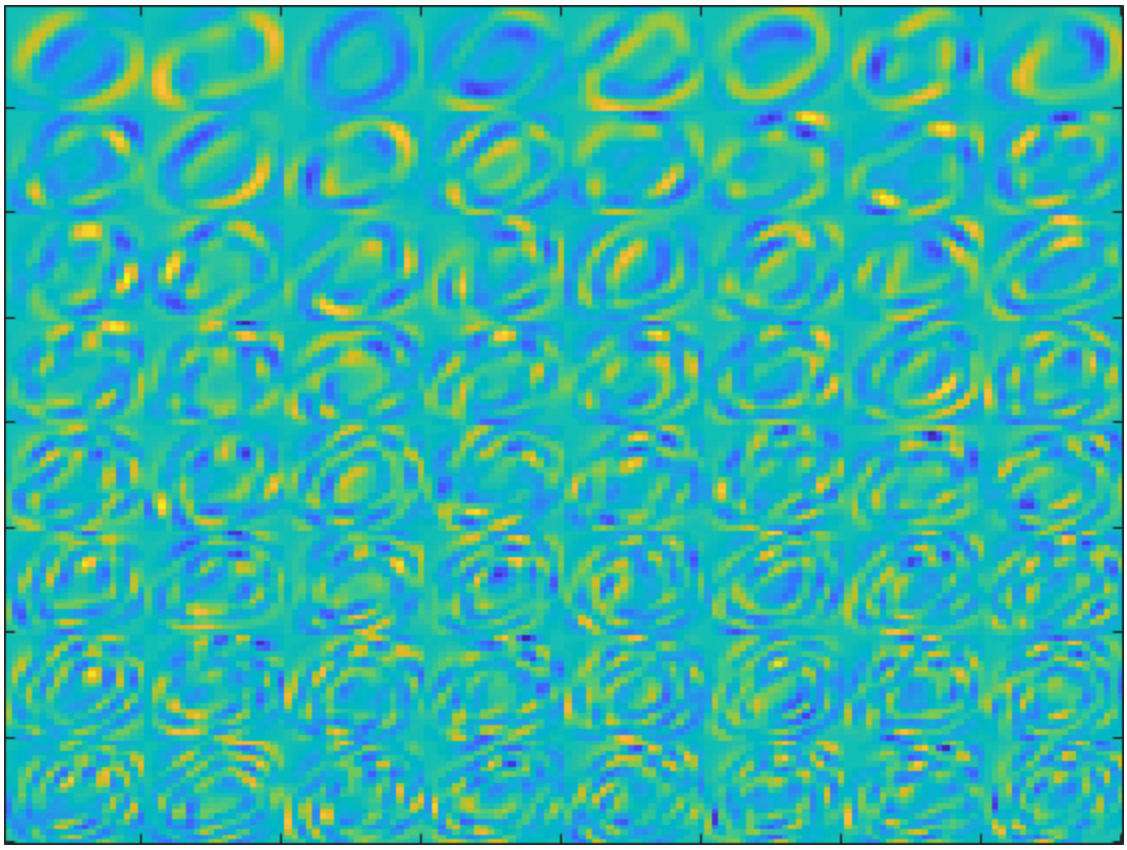
Perform SVD on digit 0, Eigenfaces, Reconstruction

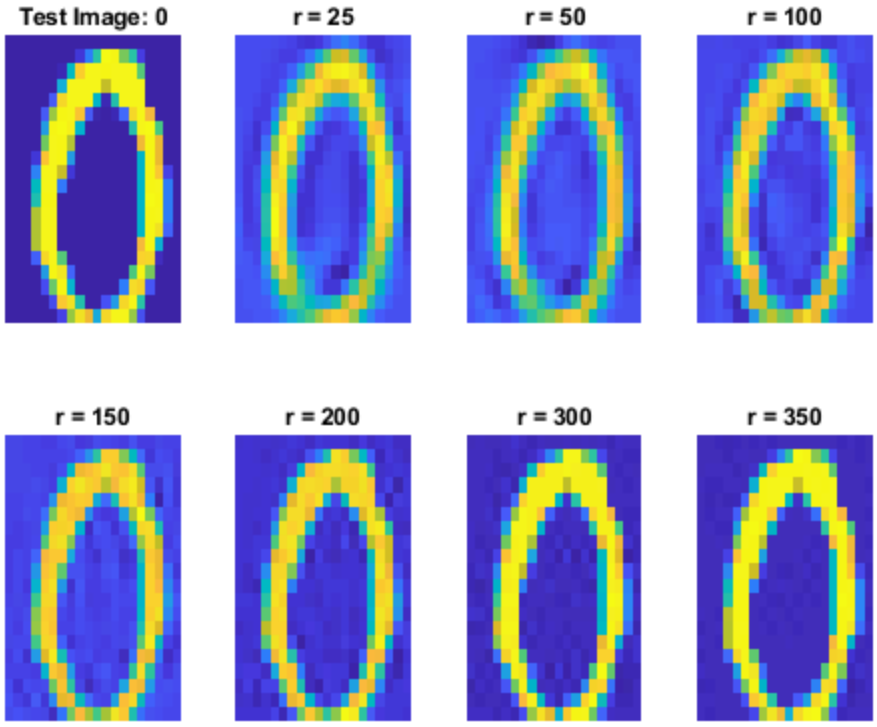
```
%Perform SVD on the mean of the digit 0
[U,S,V]= svd(database_trainImg{MS,1},'econ');
```

Plot Eigen Faces

```
EigenFaces = zeros(20*8,20*8);
count = 1;
for i = 1:8
    for j = 1:8
        EigenFaces(1+(i-1)*n:i*n,1+(j-1)*m:j*m)...
            = reshape(U(:,count),[20,20]);
        count = count +1;
    end
end
figure(2), axes('position',[0 0 1 1]),axis off
imagesc(EigenFaces)

% Reconstruction of digit 0
figure(3)
subplot(2,4,1)
testDigit = database_testImg{img_sort,1}(:,3);
```

```
digitAvg = database_trainImg{avg_img,1};
testDigitMS = testDigit - digitAvg;
imagesc(reshape(testDigit,[20,20])), axis off
title('Test Image: 0');
count = 1;
% Determining Rank r
for r = [25 50 100 150 200 300 350]
    count = count+1;
    subplot(2,4,count)
    reconDigit = digitAvg + (U(:,1:r)*(U(:,1:r)'*testDigitMS));
    imagesc(reshape(reconDigit,[n,m])),axis off
    title(['r = ',num2str(r,'%d')]);
end
```

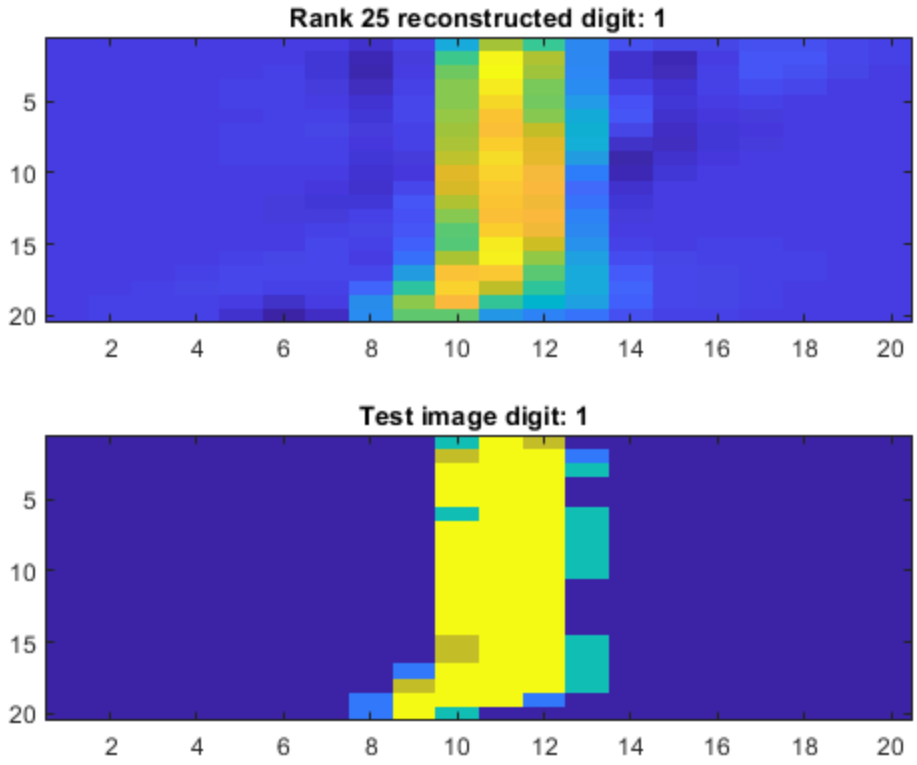




Euclidean distance Between Numbers, r = 25

```
[U,S,V]= svd(database_trainImg{MS,2},'econ');
r = 25;
testDigit = database_testImg{img_sort,2}(:,3);
digitAvg = database_trainImg{avg_img,2};
testDigitMS = testDigit - digitAvg;

reconDigit = digitAvg + (U(:,1:r)*(U(:,1:r)'*testDigitMS));

figure(4)
subplot(2,1,1);

imagesc(reshape(reconDigit,[n,m]))
title('Rank 25 reconstructed digit: 1');

subplot(2,1,2);
imagesc(reshape(testDigit,[20,20]))
title('Test image digit: 1 ');

E_dist = sqrt(sum(((testDigit - reconDigit).^2)));
disp(['Euclidean Distance = ',num2str(E_dist)])
```

Euclidean Distance = 1.7666



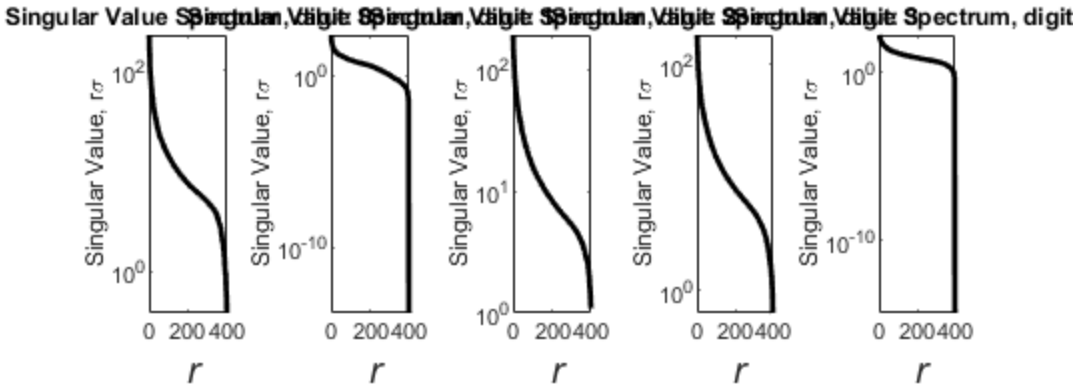
All Questions and Performances

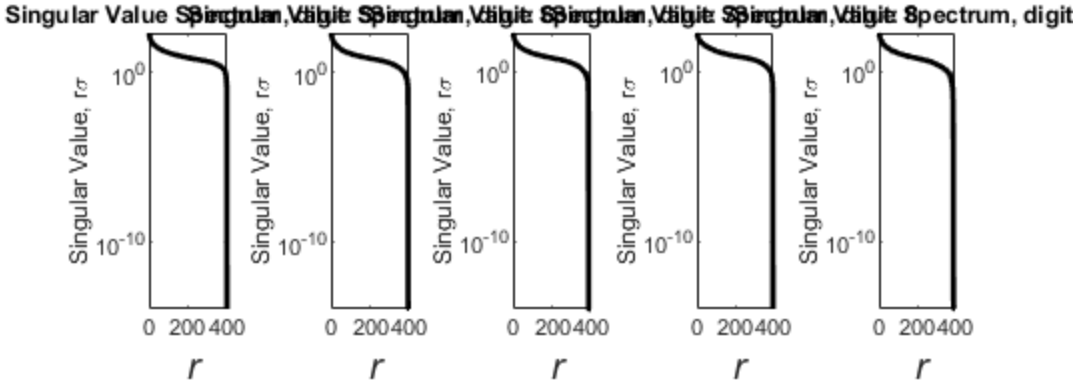
Perform SVD analysis for each number (0 to 9) 1. Interpretation of U, Sigma, V matrices: X = U*Sigma*V' U: - Contains information on column space of X which contain the features of individual samples. Sigma: - Diagnal matrix that determines how important the columns of U and V are in a hierarchical manner. V: Contains information on the row space of X, doesn't seem important for this assigment.

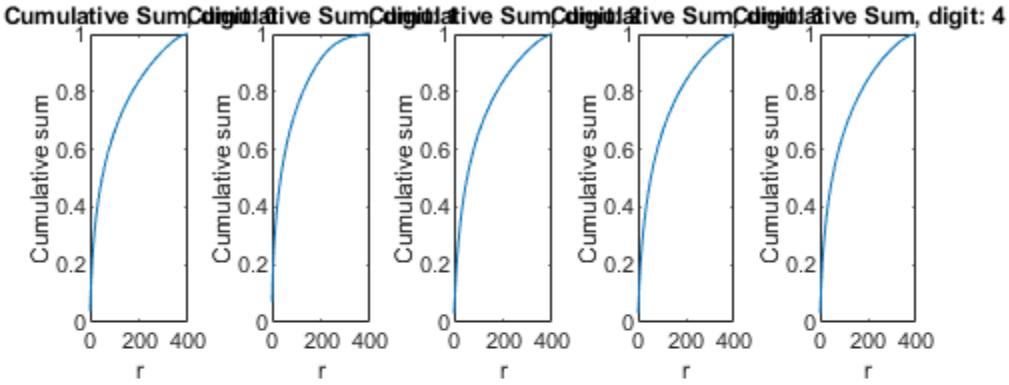
```
r = 25;
disp(['rank r = ', num2str(r)]);
for i = 1:10

[U,S,V]= svd(database_trainImg{MS,i},'econ');

% 2. Singular value spectrum
figure(5)
subplot(2,5,i)
semilogy(diag(S),'k','linewidth',2);
title(['Singular Value Spectrum, digit: ', num2str(i-1)]);
xlabel('\it r','fontsize',14);
ylabel('Singular Value, r\sigma');
% 3. Determine rank r for good image reconstructions
figure(6)
subplot(2,5,i)
plot(cumsum(diag(S)/sum(diag(S))), '-');
title(['Cumulative Sum, digit: ', num2str(i-1)]);
xlabel('r');
```

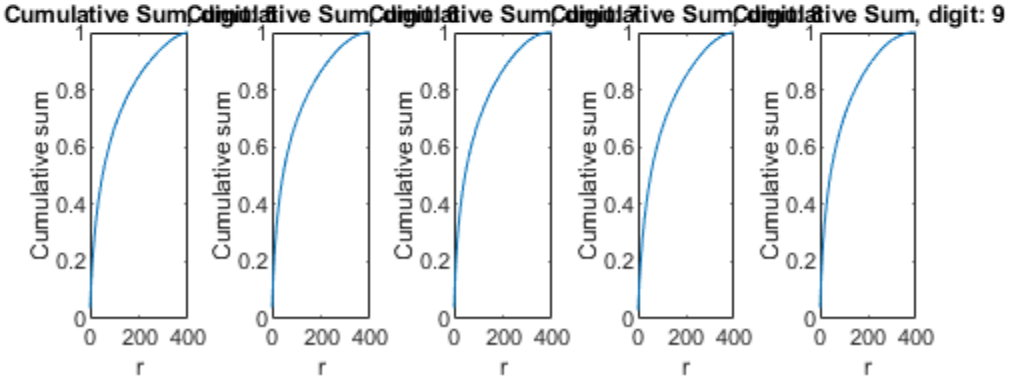
```
ylabel('Cumulative sum');

% 4. Compare differences between images of the same digit
% random test image
testDigit = database_testImg{img_sort,i}(:,randi([1 100]));
digitAvg = database_trainImg{avg_img,i};
testDigitMS = testDigit - digitAvg;
reconDigit = digitAvg + (U(:,1:r)*(U(:,1:r)'*testDigitMS));
E_dist1 = sqrt(sum(((testDigit - reconDigit).^2)));
disp(['Euclidean Distance (same)digit: '...
    , num2str(i-1), ': ', num2str(E_dist1)]);

% 5. Compare differences between images of the different digit
% random test image
v = randi([1 10]);
testDigit = database_testImg{img_sort,v}(:,randi([1 100]));
digitAvg = database_trainImg{avg_img,i};
testDigitMS = testDigit - digitAvg;
reconDigit = digitAvg + (U(:,1:r)*(U(:,1:r)'*testDigitMS));
E_dist2 = sqrt(sum(((testDigit - reconDigit).^2)));
disp(['Euclidean Distance: ', num2str(v),' and ',...
    num2str(i-1), ' : ', num2str(E_dist2)]);
end
```

```
rank r = 25
Euclidean Distance (same)digit: 0: 2.3912
Euclidean Distance: 1 and 0 : 3.3822
Euclidean Distance (same)digit: 1: 1.258
Euclidean Distance: 10 and 1 : 6.5096
Euclidean Distance (same)digit: 2: 3.5892
Euclidean Distance: 6 and 2 : 4.4606
Euclidean Distance (same)digit: 3: 4.6264
Euclidean Distance: 2 and 3 : 4.1481
Euclidean Distance (same)digit: 4: 3.3278
Euclidean Distance: 4 and 4 : 5.7442
Euclidean Distance (same)digit: 5: 3.1676
Euclidean Distance: 5 and 5 : 5.1172
Euclidean Distance (same)digit: 6: 3.7589
Euclidean Distance: 9 and 6 : 5.1397
Euclidean Distance (same)digit: 7: 2.3239
Euclidean Distance: 4 and 7 : 7.7033
Euclidean Distance (same)digit: 8: 3.0277
Euclidean Distance: 2 and 8 : 3.3853
Euclidean Distance (same)digit: 9: 2.9941
Euclidean Distance: 9 and 9 : 6.129
```


Singular Value Spectrum, digit
Singular Value, rσ
r


Singular Value Spectrum, digit
Singular Value, rσ
r


Cumulative Sum, digit: 4
Cumulative sum
r


Cumulative Sum, digit: 9
Cumulative sum
r

Published with MATLAB® R2021a