Project Assignment #1

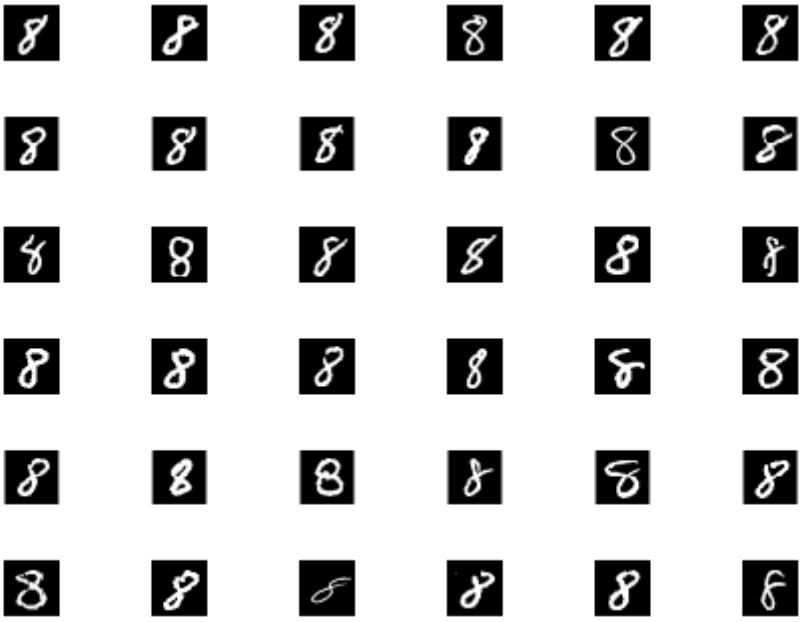
Contents

Get started

The MNIST databse of handwritten digits -- Yann LeCun (NYU) Corinna Cortes (Google), Chris J. C. Burges (Microsoft Research)
download mnistdata.mat (13MB) and viewdigit.m

```
clear;
load mnistdata;

% Visualize a selected train/test digit

figure(1)
n = 6;
for i = 1:n*n

   digit = train8(i,:);
   %digit = test8(i,:);

   digitImage = reshape(digit,28,28);

   subplot(n,n,i);
   image(rot90(flipud(digitImage),-1));
   colormap(gray(256));
   axis square tight off;

end

% Visualize the average train digits

T(1,:) = mean(train0);
T(2,:) = mean(train1);
T(3,:) = mean(train2);
T(4,:) = mean(train3);
T(5,:) = mean(train4);
T(6,:) = mean(train5);
T(7,:) = mean(train6);
T(8,:) = mean(train7);
T(9,:) = mean(train8);
T(10,:) = mean(train9);

for i = 1:10
```

```
    digitImage_mean(:,:,i) = reshape(T(i,:),28,28);
end

figure(2)
for i = 1:10
    subplot(2,5,i)
    image(rot90(flipud(digitImage_mean(:,:,i)),-1));
    colormap(gray(256));
    axis square tight off;
end
```



The first figure above gives a sample of different instances of digit 8 in the data matrix 'train8'. The second figure shows the average of all the 0 digits, the average of all the 1 digits, and so on.

The data is separated into two categories: train and test. Each category contains 10 sets of digits from 0-9. Each row vector of length 784 is a 28-by-28 image.

```
whos('-file','mnistdata.mat')
```

```
  Name           Size               Bytes  Class    Attributes

  test0        980x784            768320  uint8
  test1       1135x784            889840  uint8
  test2       1032x784            809088  uint8
  test3       1010x784            791840  uint8
  test4        982x784            769888  uint8
  test5        892x784            699328  uint8
  test6        958x784            751072  uint8
  test7       1028x784            805952  uint8
  test8        974x784            763616  uint8
  test9       1009x784            791056  uint8
  train0      5923x784           4643632  uint8
  train1      6742x784           5285728  uint8
  train2      5958x784           4671072  uint8
  train3      6131x784           4806704  uint8
  train4      5842x784           4580128  uint8
  train5      5421x784           4250064  uint8
  train6      5918x784           4639712  uint8
  train7      6265x784           4911760  uint8
  train8      5851x784           4587184  uint8
  train9      5949x784           4664016  uint8
```

In this project, we investigate two ways to classify a digit: centroid algorithm and PCA algorithm.

Part 1: The Centroid method

Step a: Calculate the distance between the means and the test image

The following code takes a digit from the testing set and computes the 2-norm distances between this digit and the 10 average train digits computed above.

```
z = double(test7(55,:));
dist = zeros(10,1);
for k=1:10
    dist(k) = norm( z - T(k,:) );
end
dist
```

```
dist =

     2.409921404064263e+03
     2.079022907553916e+03
     2.098626637203944e+03
     1.952902851210245e+03
     1.980993001946881e+03
     1.968942601296248e+03
     2.193676576493703e+03
     1.564492873576325e+03
     1.933083588042667e+03
     1.669572709954699e+03
```

Since the 2-norm distance between unknown digit and the average train digit 7 is smallest, our simple classification algorithm will label unknown digit as '7' which is indeed the correct answer.

Note also that average train digit 7 is actually T(8,:) because MATLAB indexing starts at 1.

Step b: Classify the test set of digit

Write a function that takes as inputs:

- an n-by-784 matrix A containing n digits
- a 10-by-784 matrix T containing the average train digits

and ouputs:

- an n-by-1 vector containing the labels (0-9) for digits in A.

Step c: Report the success rate (correct/total) of each digit

Classify the entire test set and report the success rate of each digit.

Part 2: The PCA Method

The PCA method attempts to identify characteristic properties of each digit, based on the training data, and compares these properties with those of the test digit in order to make an identification. Here, the characteristic properties are the principal components extracted from the training set.

Step a: Find the principal components of the training set

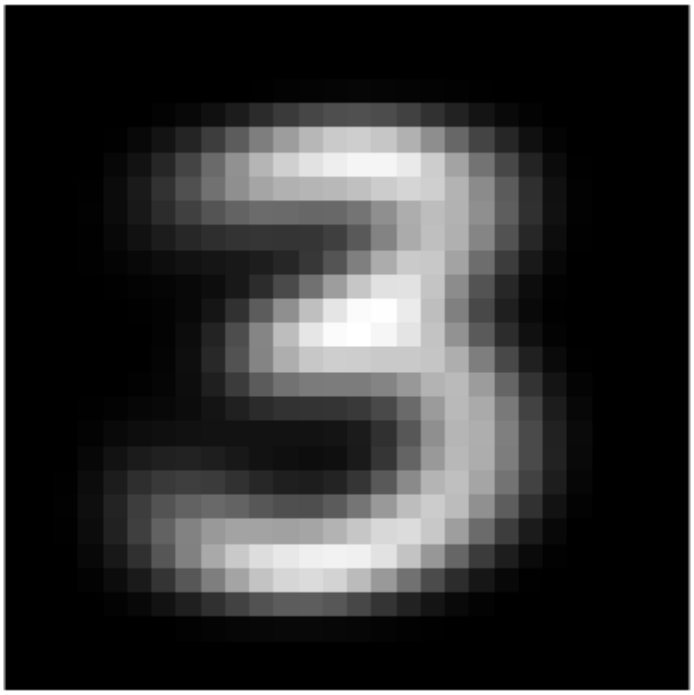
Find the first 5 singular vectors of the transpose of train3.

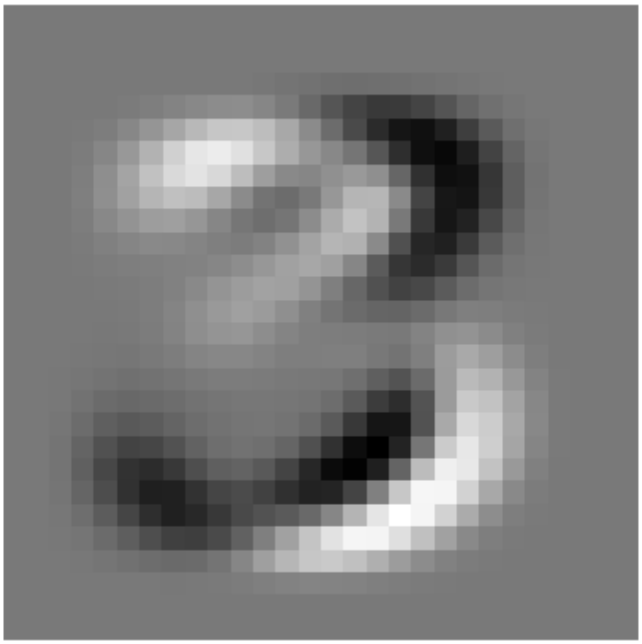
```
[U3,~,~] = svds(double(train3'), 5) ;
size(U3)
```

```
ans =

   784     5
```

These five singular vectors represent the five dominant chracteristics of the digit '3'. They also form an approximate basis for the space of all possible digit 3's.

```
viewdigit( U3(:,1) );

viewdigit( U3(:,2) ) ;
```





We can test how well a digit z can be represented in this basis U_3 of digit 3 by solving the least square problem:

$$\min_x \|z - U_3 x\|_2$$

which has the solution:

$$\min_x ||z - U_3 x||_2 = ||z - U_3 U_3^T z||_2$$

Now that we have a way to measure how far away an unknown digit is from 'looking' like digit 3, let's do the same for all digits 0 to 9. We generate approximate bases for all ten digits in this way:

```
basis_len = 5;
Us=zeros( 28*28, basis_len, 10);
for k=1:10
    % go through each digit 0 to 9
    s = strcat('train',num2str(k-1));
    A = double(eval(s));

    % and get first 5 singular vector
        [U,~,~] = svds( A', basis_len );
    Us(:,:,k)=U;
end
```

The essence of this PCA approach is that for a given unknown digit z (say, test4(15,:)) we compute this number

$$||z - U_k U_k^T z||_2$$

for all digit k from 0 to 9 and choose z's label based on how well z are represented by approximate bases $U_0, U_1, \ldots, U_9$. We give z the label 3 for example if

$$||z - U_3 U_3^T z||_2$$

gives the smallest number. Let's try that on a test digit:

```
z = double(test4(14,:))';
dist = zeros(10,1);
for k=1:10
    Uk = Us(:,:,k);
    dist(k) = norm( z - Uk*(Uk'*z) );
end
dist
```

```
dist =

     2.185939704715336e+03
     2.171967417549981e+03
     2.130547447175397e+03
     2.157261611565722e+03
     1.487302672264204e+03
     2.120820230633459e+03
     2.025370293771379e+03
     1.982423650244960e+03
     1.961783222042929e+03
     1.731556697795859e+03
```

Since test4(14,:) is best represented by the approximate basis of digit 4 (this is indicated by the smallest number). It gets labeled with '4' which is the correct answer!

Write a function that takes as inputs:

- an n-by-784 matrix A containing n digits
- a 784-by-k-by-10 matrix T containing the first k singular vectors for each of the training sets (train0',train1',train2'...,train9') (tranposed!)

and ouputs:

- n-by-1 vector containing the labels (0-9) for digits in A.

Step b: Test and report the success rate

- As in step b of Part 1, test and report the success rate of this PCA approach
- Experiment with larger approximate bases (i.e. increase the basis_len).
- (Optional) Try bases of different lengths for different digits

---

Published with MATLAB® R2017a