PCA for clustering MNIST images

Clear workspace and close figures

```
close all;
clearvars;
```

Load data and reshape

```
load('digitsPCA.mat');
[h,w,numImages]=size(Data);
numPixels=h*w;
X = reshape(Data,numPixels,numImages);
```

Labels for images

```
% Image labels
labels= zeros(1, numImages);
labels(1:500)=1;
labels(501:end)=7;
```

Mean subtraction (centering) and pixel covariance matrix

```
% Mean of each pixel
mu = mean(X,2);

% Normalization by mean subtraction
X_centered = bsxfun(@minus, X, mu);
```

If you don't center the data, the first eigenvector (i.e the one with largest eigenvalue) will be roughly collinear to the mean vector.

Since PCA is constrained to give an orthogonal basis set, all other directions will be orthogonal to teh mean vector. These are not the directions

of maximum variance. Removing the mean gives the correct covariance and eigendecomposition.

```
% Q is the covariance Matrix numPixels X numPixels
Q = X_centered*X_centered'/(numImages-1);
```

The size of this matrix is 784 by 784 (numPixels by numPixels). The i-i entries represent pixel variance. The i-j entries indicate covariance between pixel i and j.

SVD and reconstructing the covariance

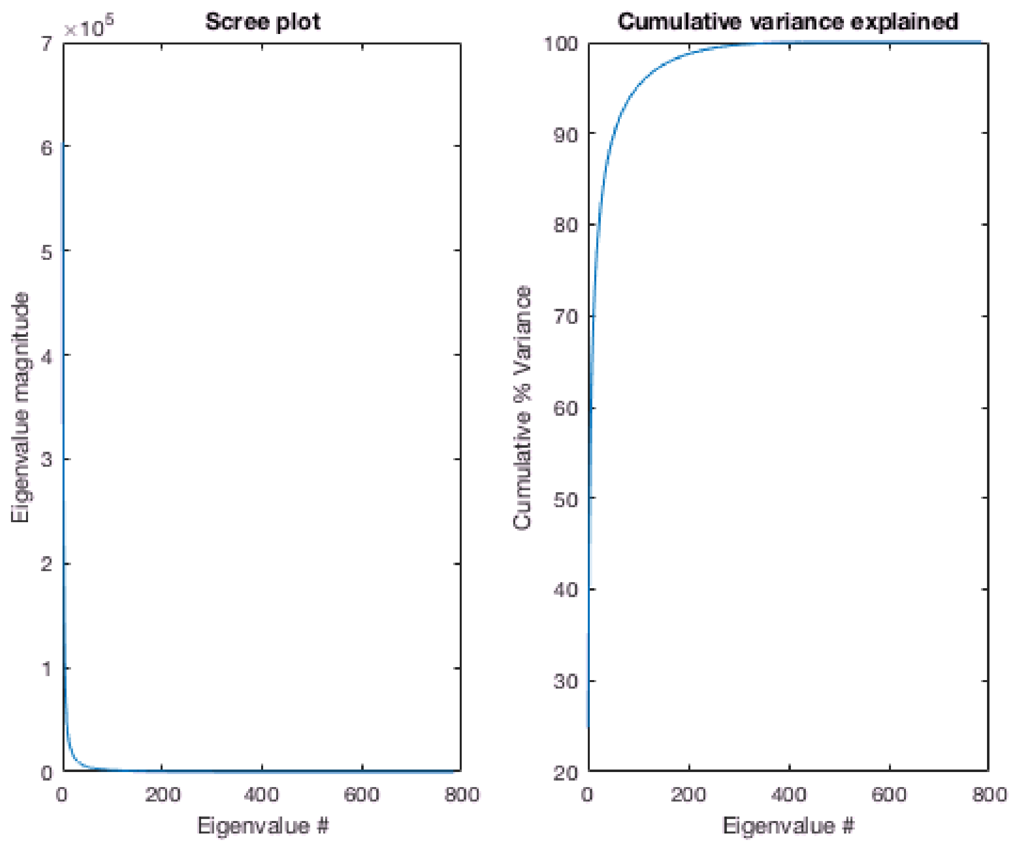
```
% SVD
[U,S,V] = svd(X_centered,'econ');
```

```
lambdas = diag(S*S'/(numImages-1)); % eigenvalues of Q
Q_SVD = U*(S*S')*U'/(numImages-1); % columns of U form an orthonormal basis set for Q
```

The columns of U are the eigenvectors of Q. These capture the diredctions of maximum variance for the data set in the 784 dimensional space. These directions are constrained to be orthogonal.

Variance explained per PC

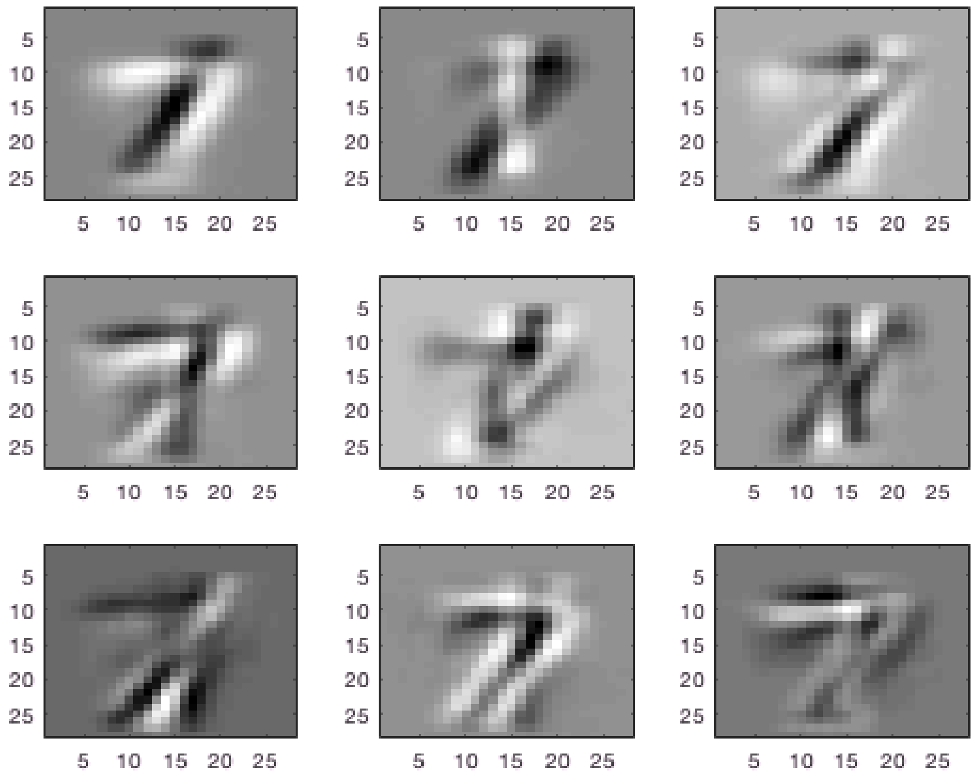
```
% Variance explained per PC
Var_expl =100 * cumsum(lambdas)./sum(lambdas);

figure(1);
subplot(1,2,1)
plot(1:numPixels, lambdas);
title('Scree plot')
xlabel('Eigenvalue #')
ylabel('Eigenvalue magnitude')

subplot(1,2,2)
plot(1:numPixels, Var_expl);
title('Cumulative variance explained')
xlabel('Eigenvalue #')
ylabel('Cumulative % Variance')
```



The first 2 components capture approxiamtely 37% of the data variance

Visualizing the eigenvectors as images

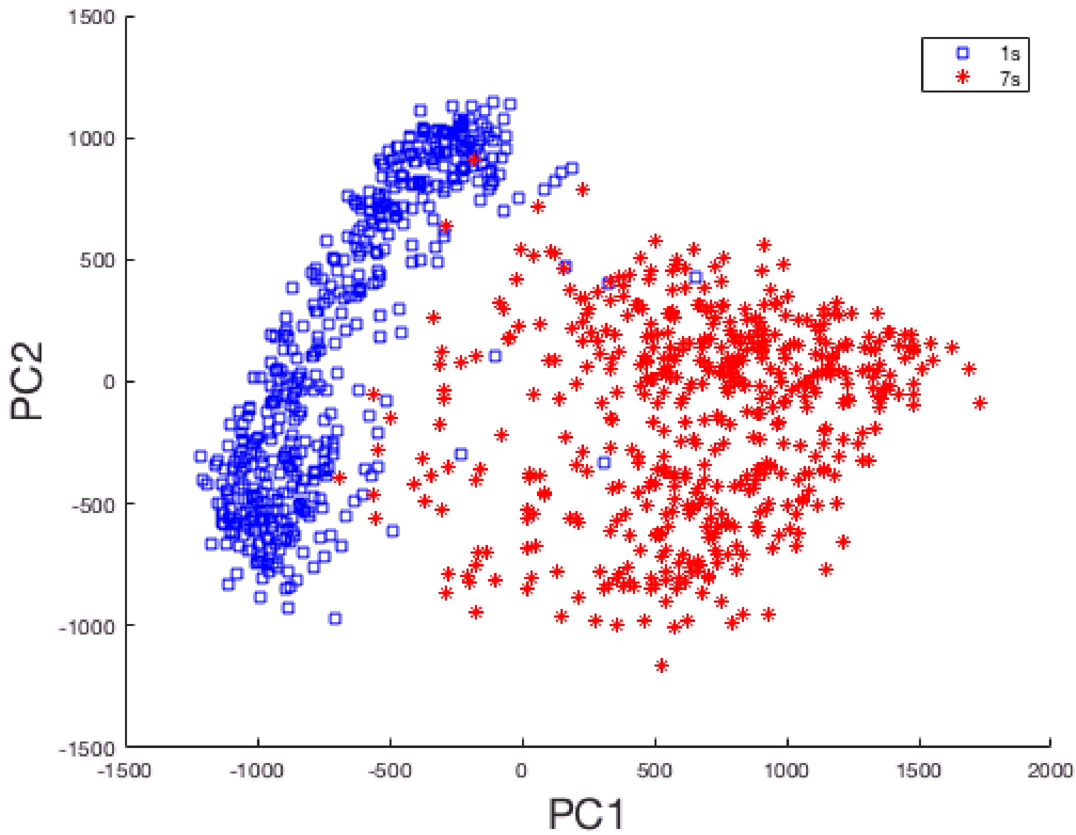
```
figure(3)
for k = 1:9
    subplot(3,3,k)
    imagesc(reshape(U(:,k),28,28));
    colormap gray;
end
```



These images represent the pixel intensity patterns or features that capture most of the variability in the image. Pixels with high intensities will in general covariate well.

Rotate mean subtracted data and visualize projections

```
Proj = U'*X_centered;
figure
plot(Proj(1,labels==1), Proj(2,labels==1), 'sb')
hold on
plot(Proj(1,labels==7), Proj(2,labels==7), '*r')
legend('1s', '7s')
xlabel('PC1', 'fontsize', 20)
ylabel('PC2', 'fontsize', 20)
%set(gcf, 'color', 'w')
%set(gca, 'fontsize', 16)
hold off
box off
```



Strictly speaking the data is not linearly separable as any line inte PC1-PC2 space will produce some misclassification.

However, a linear classifier will do remarkably well in this feature space.