Name:	
ID number:	LiU-ID:
Passed:	Date:

TSBB06 Computer Exercise E
Principal Component Analysis

Developed by Klas Nordberg.

Computer Vision Laboratory, Linköping University, Sweden

October 31, 2018

Introduction

In this exercise you will work with various aspects of principal component analysis (PCA). In the first part of the exercise you will analyze an unknown signal using PCA. In the second part you will instead analyze both natural and synthetic images.

Starting and setting things up

In order to simplify the exercises, you should write the Matlab code that you use into a script file that can be executed in order to generate all results that are requested. This script file should be placed in your own home directory, so you need to add it to your path (if it is not already there). Start Matlab and change directory to the folder that is used in these exercises, and add your own local directory to the path:

```
cd /site/edu/bb/MultidimensionalSignalAnalysis/ExerciseE/
addpath ...
```

Useful Matlab functions

`sum` computes the sum of the elements in a vector, e.g., a vector that contains the eigenvalues of $\mathbf{C}$.

cumsum computes a cumulative sum of the elements in a vector, producing a new vector.

1 Preparatory exercises

Before coming to the computer exercise it is necessary that you have a clear picture of the material related to Principal Component Analysis (PCA) (lecture 2E). Below, you will find a number of preparatory exercises to be answered **before** the session in order to save time for you, the assisting teacher, and your fellow students. You need to read and understand the specific tasks of the rest of the guide in order to answer some of the preparatory exercises.

1. What is a principal component of the signal?
ANSWER:

2. What is the magnitude of a principal component?
ANSWER:

3. If a principal component has a large magnitude and another has a low magnitude, what is the difference between the two relative to the signal?
ANSWER:

4. In Section 2 you will have to figure out how many principal components that are reasonable to use for some unknown signal, by analyzing the differences of the magnitudes between one principal component and the following. What is the motivation for this approach?
ANSWER:

5. How is the cost function ϵ defined for PCA? What undetermined entity is it that you minimize ϵ over?

ANSWER:

6. In the case that the optimal subspace has dimension k, i.e., the number of principal components is k, what is the minimal value of ϵ?

ANSWER:

7. In Section 3.3 you will apply PCA to small image blocks of uncorrelated random pixels. What type of distribution of the eigenvalues in $\mathbf{C}$ do you expect to get in this case?

ANSWER:

2 PCA of an unknown dataset

In this task you are given samples of an unknown dataset $\mathbf{v}$ that you will analyze in terms of its statistical properties. The dataset is represented in terms of a data matrix $\mathbf{A}$ for which each column is a sample of the signal. The signal $\mathbf{v}$ is 100-dimensional and the data matrix contains 600 samples of this signal. Begin by loading the data matrix and check its size:

```
load A
size(A)
```

Compute the principal components of the dataset. This can be done either by computing an eigenvalue decomposition (EVD) of the correlation matrix $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ or a singular value decomposition (SVD) of the data matrix $\mathbf{A}$. The first approach has the advantage of using the well-known EVD but has the disadvantage of providing, in some cases, a lesser numerical accuracy and also that some numerical tools, such as Matlab, computes eigenvalues without sorting them in any particular way. Conversely, the second approach has the advantage of better numerical accuracy and also providing sorted singular values by the mathematical tools only at the expense of understanding the basic properties of the SVD. In this computer exercise you can choose whichever approach you feel it more suitable to you.

In the first approach, based on EVD, the principal components are computed as:

```
C = A*A';                          %Compute the correlation matrix C
[e l] = eig(C);                    %Compute EVD of C
[PM p] = sort(diag(l),'decend');   %Sort the eigenvalues: largest first
PC = e(:,p);                       %Sort the eigenvectors in the same way
```

In the second approach, based on SVD, the principal components are computed as:

```
[PC S] = svd(A);                   %Compute SVD of A
PM = diag(S);                      %magnitudes are given by the
                                   %singular values
```

In both cases, the principal components form an ON-basis that ends up as columns in the $N \times N$ matrix $\mathbf{PC}$ and their corresponding magnitudes are in the N-dimensional vector $\mathbf{PM}$, sorted with largest magnitude in the first element. Note that in the SVD case, the right singular vectors are never used.

Apply either approach for computing the principal components and their

magnitudes and plot the magnitudes/eigenvalues/singular values as well as the logarithms of the latter:

```
PM = ...
PC = ...
figure(1);
subplot(2,1,1):plot(PM,'o');
subplot(2,1,2):plot(log(PM),'o');
```

QUESTION: How many of the principal components M do you consider to be *significant* for representing the signal? Why?
ANSWER:

The statistical analysis of an unknown signal is not straight-forward since it is its statistical properties that are not known. Therefore several strategies can be considered. One approach to determine the number of significant principal components is to look at the difference in the magnitudes and include the largest principal components that make a significant difference in the error.

Try different values for M and choose the one that you believe is reasonable based on this analysis.

```
M = ...
figure(2);plot(diff(PM((M + 1):end)),'o');
```

QUESTION: What value did you choose for M? Was it the same as before?
ANSWER:

One way to understand the signal's statistical properties is to project the signal onto a pair of principal components and plot the corresponding coordinates. Do this for the first two principal components:

```
figure(3);plot(PC(:,1)'*A,PC(:,2)'*A,'o');axis('equal');
```

QUESTION: What can you say about the statistical properties of the signal from this view, e.g., its distribution along the two principal components?

ANSWER:

Do a similar analysis also for the case when the mean of the signal subtracted before the principal components are computed:

```
m=mean(A,2);
A0 = A - m*ones(1,size(A,2));
PC = ...
PM = ...
figure(4);
subplot(2,1,1);plot(PM,'o');
subplot(2,1,2);plot(log(PM),'o');
figure(5);
subplot(4,1,1);plot(PC(:,1)'*A0,PC(:,2)'*A0,'o');axis('equal')
subplot(4,1,2);plot(PC(:,2)'*A0,PC(:,3)'*A0,'o');axis('equal')
subplot(4,1,3);plot(PC(:,3)'*A0,PC(:,4)'*A0,'o');axis('equal')
subplot(4,1,4);plot(PC(:,4)'*A0,PC(:,5)'*A0,'o');axis('equal')
```

These last plots show you the projection of the signal onto principal components (1,2), (2,3), (3,4) and (4,5). Another way to investigate the signal is to look at the error ϵ that is minimized by the principal component analysis.

Compute and plot the residual error ϵ as a function of the number of principal components that you use. Hint: use the Matlab function `cumsum`. **Remember that the eigenvalues of C are the squares of the singular values of A!**

```
figure(6);plot(...,'o');
```

QUESTION: How many principal components do you believe to be significant in this case? Why?

ANSWER:

QUESTION: Is it the same number as before, when the mean was not subtracted? Why?

ANSWER:

Try plotting the projection of the data onto the three largest principal components with Matlab's command `plot3` command to get a better feeling for the distribution of points. Use `axis('equal')` to assure equal scaling of the axes:

```
figure(7);plot3(...);axis('equal');
```

QUESTION: Given the observations that you have made during the analysis of this signal, how would you describe its distribution?

ANSWER:

QUESTION: Try to suggest some other strategy for determining the number of significant principal components M.

ANSWER:

QUESTION: Plot the data matrix `A0` using `mesh` or `imagesc`. Is it possible to make the same observation from these plots as you did using `plot3`?

ANSWER:

3 Principal component analysis of images

In this task you will analyze a signal that is generated by considering smaller regions of an image. There are several natural images of format PNG in the directory of this exercise. View them first in some external image viewer, or load them into Matlab and display them. Finally, select one that you want to analyze (do not use the synthetic image *ploop*). The analysis implies that a data matrix is formed from distinct blocks of size $N \times N$ pixels in the image, a suitable number of principal components are chosen, and the blocks are projected onto the corresponding subspace and reshaped into an image. In this task, the analysis is made without subtracting the mean of the signal.

Load the image and display it:

```
im = double(imread('middlebury.png')); % Choose you image here!
size(im)
figure(10);colormap('gray');imagesc(im);
```

Form the data matrix from non-overlapping 8×8 regions in the image:

```
N = 8;
A = im2col(im,[N N],'distinct');
size(A)
```

Compute the principal components and plot their magnitudes:

```
PC = ...
PM = ...
figure(11);
subplot(2,1,1);plot(PM,'o');
subplot(2,1,2);plot(log(PM),'o');
```

For $M = 1, \ldots, 6$ compute the coordinates of each block (the columns of $\mathbf{A}$) relative to the corresponding principal component basis, reconstruct the block with the basis and reshape all blocks into an image that is displayed:

```
figure(12);colormap('gray');
for M=1:6,
  c=PC(:,1:M)'*A;    %Compute coordinates from blocks
  Arec=PC(:,1:M)*c;  %Reconstruct blocks from coordinates
  imrec=col2im(Arec,[N N],size(im),'distinct');  %Reshape into image
  subplot(2,3,M);imagesc(imrec);axis('off');     %Display image
  title(sprintf('%d principal components',M));   %Set title
end
```

This reconstructs the image blocks by means of up to 6 principal components. This may or may not be sufficient for a particular image, there is nothing that suggests that images in general can be represented by 6 principal components.

QUESTION: Choose at least 4 different images and investigate how many principal components you would like to use to make a reasonable representation of the image. Which images and how many components?
ANSWER:

QUESTION: In which part of an image do you obtain a good representation of the original image, and where is it not as good?
ANSWER:

QUESTION: The image *middlebury* contains a white box with some text on it. How many principal components are needed in order to represent an image where you can read the text? Is this number something can you can derive from the distribution of principal magnitudes?
ANSWER:

3.1 The principal components

Display the 6 largest principal components as $N \times N$ blocks:

```
figure(14);colormap('gray');
for cnt=1:6,
  subplot(2,3,cnt);h=mesh(reshape(PC(:,cnt),N,N));
  set(h,'edgecolor','black');axis([1 N 1 N -0.5 0.5]);
title(sprintf('principal component %d',cnt));
end
```

QUESTION: Can you characterize the principal components in some simple way?
ANSWER:

Compute the principal components from one image, and reconstruct another image using the 6 largest principal components from the first image.

QUESTION: Is there a significant difference in the quality of the reconstruction compared to when the principal components comes from the same image? Why?
ANSWER:

3.2 Changing the block size

Repeat the same type of calculations as above, but now with a smaller and a larger size of the blocks, e.g., 4×4 and 16×16.

QUESTION: What is the significant difference when the block size changes? Why?
ANSWER:

QUESTION: How many components do you need for block size 16×16 in order to get the same quality in the reconstruction as with 6 components with block size 8×8? Explain why.
ANSWER:

3.3 PCA of synthetic images

Do the same type of computations on the synthetic image *ploop*. Plots its principal values and largest principal components.

QUESTION: Is there any qualitative difference between the principal components and principal values derived from this image compared to those derived from natural images?
ANSWER:

In a similar way as above, use the principal components computed from this synthetic image and reconstruct the blocks from a natural image.

QUESTION: Is the reconstruction reasonably good? Explain why.
ANSWER:

Define a random image:

```
im = rand(512,512);
```

and do the same computations and plots as before. In particular, compute and plot the principal values and principal components for this case.

QUESTION: Do they appear as your answer to preparatory exercise 7?
ANSWER:

Compute a small number of principal components from the noise image and use them to reconstruct on a natural image, in the same way as before.

QUESTION: What is the result?
ANSWER:

Use a PCA basis computed from a natural image and use it to reconstruct the noise image.

QUESTION: What is the result? How is it different compared to the original noise image?
ANSWER: