Exercises 8: PCA

Introduction to data science and Python programming for medical research (2025)

If you have not done so already you need to install the sklearn package. For windows users this would require you to open your conda terminal, found in the start menu, and then typing:

```
conda install sklearn
```

For this exercise you will load some image data that comes in the medical imaging DICOM format and you will need two additional packages for this purpose.

```
conda install pydicom
conda install SimpleITK
```

1 PCA for visualization

Recreate one of the PCA plots from the lecture.

Use either the breast cancer or the diabetes dataset, apply PCA and plot the transformed data points in 2D.

What do these two principal component projections of the data we have plotted mean? Can we give an interpretation?

Optional: Use the function to remove numbers and ticks on the axes provided for the next exercise. PCA plots should never have numbers on the axes!

Optional: Label the axes and indicate the fraction of the explained variance along each.

2 Respiratory motion PCA analysis and modeling of cine-MR images

Background

Respiratory motion needs to be considered when giving patients with lung cancer radiation therapy. Several imaging techniques can be used to determine and model the extent of tumor displacement during breathing.

One such imaging technique is cinematic mode magnetic resonance (cine-MR) which gives good quality images with high temporal resolution. However, the images are only 2D.

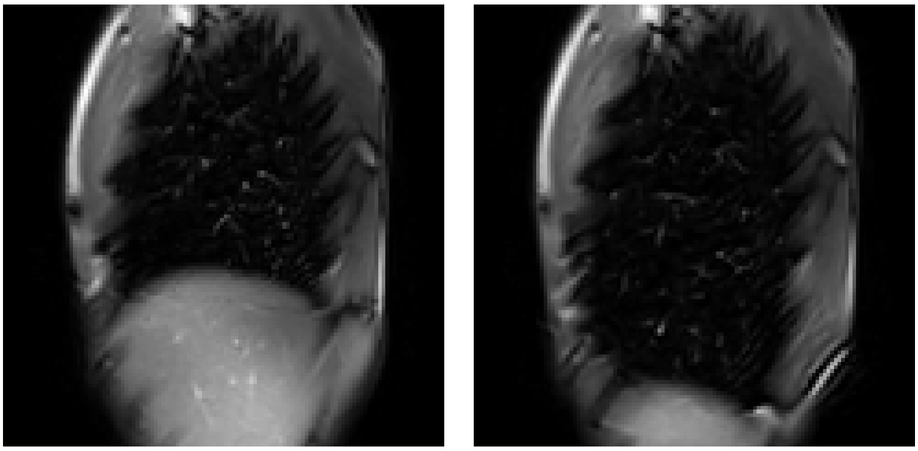

Figure 1: Two cine-MR images from the sequence of images. On the left is an image taken during exhale and on the right an image taken during inhale.

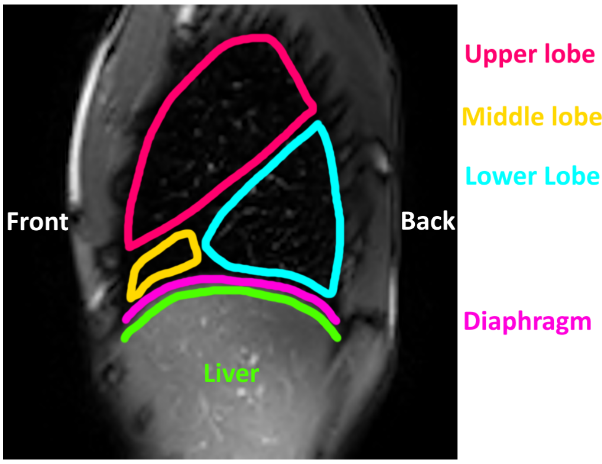

Figure 2: The cine-MR images were acquired in a sagittal plane such that the lung and the liver is visible.

Breathing is characterized by a motion along one dominating direction, up and down as a person breathes in and out. As such we hypothesize that a PCA model should be able to capture this in its first principal components.

Task

The image set you will be using are a sequence of 450 images of a healthy test subject breathing normally. The images are taken in a sagittal plane where the lung, diaphragm and liver can be seen.

For this exercise you are provided with a script you can use as basis for your answers as well as a script containing two support methods for loading the data and removing ticks and numbers from the axes of a plot. These instructions are written in the script intended for the answers as well.

2.1 Load images

Specify directory and load images. The loaded images will be a list of 2D numpy arrays. Each entry in the list represents one image.

2.2 Flatten images and construct the data matrix

Flatten all images and use them to construct a data matrix.

Store the shape of the original images in a variable. You will need it later.

Tip: Use a for-loop to iterate over all images in the list. The flattened images can be appended to another list and this list can be used to create a 2D numpy array, the data matrix X.

2.3 Verify size of the data matrix

Print the shape of your data matrix. It should have the shape: $450 \times 16\,384$.

2.4 Create a PCA model

Create a PCA model that should retain 2 components. `fit` the PCA model to and `transform` your data. Print the shape of your transformed data. Does it look reasonable?

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA

2.5 Plot the 2D PCA transformed data

Create a figure and plot the transformed data using a scatter plot.

Since it is a PCA plot you should remove the numbers from the axes. There is a function for this in the same file as the image loading. Add labels to the two axes to indicate that they are the first and second principal components.

Also try to add the relative amount of variance they represent. Look for examples in the lecture slides.

What do you think the plot represents?

2.6 Gaussian mixture model of the transformed data

Create a GaussianMixture (GM) of the transformed data. Ie, the GM will model the data in the two PCA dimensions.

Look at the previous plot and use a reasonable number of components for the GM. Each component in this case represents one Gaussian function to be ”placed on the data” in order to model its distribution.

https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

2.7 Gaussian mixture model sampling

`sample` from your GaussianMixture to generate 10-20 new data points. You can look at the example in the lecture 7 slides on clustering. Plot the original transformed data points and these new generated data points. Use different colors for them.

Note: We are only interested in the data points `X_gm` themselves, not their labels, `y_gm`, which is also output by the `sample(...)`-method of the `GaussianMixture` class.

2.8 Reconstruct images using `inverse_transform` and plot them

Plot the PCA reconstructed images from the GM sampled data on the same plot as the transformed data points. Make sure the data matrix containing the GM samples is called `X_gm`.

`inverse_transform` the GM data point and reshape such that the flat vector is unpacked into an image with the same shape as the original images.

2.9 Plot the final result

Save or show the last plot containing the data points as well as the reconstructed images.