Statistical Signal Processing
A.A. 2017/2018

Computer Lab 3 – Principal component analysis

Duration: 3 hours

Introduction:

Hyperspectral images are scientific images of the Earth, acquired by satellites or aicrafts; rather than having three R/G/B color channels, these images have a lot more "color" components obtained through a fine sampling of the wavelength (hence the name "hyper"-spectral). The resulting 3-dimensional dataset has one image (spectral band or "color") for every sampled wavelength, which represents the measured radiance from each pixel at that specific wavelength. Hyperspectral images are very useful for image analysis. For every pixel at a given spatial position, it is possible to extract a so-called spectral vector, i.e. the 1-dimensional vector of values assumed by that pixel at all wavelengths. Assuming that each pixel is composed of just one substance, the spectral vector represents the radiance of that substance at all the wavelengths that have been sampled. Spectral vectors, therefore, can be used to infer which substance is contained in a given pixel – a typical classification problem that has a lot of practical applications in agriculture, analysis of land use / land cover, and other applications related to the study of the environment.

In this lab you will use a real hyperspectral image that has been acquired by the AVIRIS instrument, an airborne hyperspectral imager operated by the NASA. The image represents a scene of Indian Pines (Indiana, USA). It has a size of 145x145 pixels and 220 spectral bands. Along with the image, a ground truth is available, in terms of labels specifying which class (out of 16) each pixel belongs to. The classes are reported below; for more information, please see http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Indian_Pines

#	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

The purpose of this computer lab is twofold:

- To apply PCA to the spectral vectors in order to reduce their dimensionality.
- To perform classification on the reduced data (optional)

Exercise 1 – PCA

In this exercise, you will employ the Indian Pines dataset. You will not do this for the entire dataset, but only for the spectral vectors belonging to **two classes** (as in the optional exercise you will perform 2-class classification on the PCA coefficients).

Reminder: the input to the PCA must always have zero mean: besides the sample covariance, you will have to compute the **mean value $\boldsymbol{\mu}$** over the training set and subtract it from each test vector before applying PCA.

Task: You have to reduce the dimensionality of the spectral vectors of the two classes you have chosen using PCA. In particular, you should perform the following:

- Extract spectral vectors of two classes, as described above (see sample code below).
- Estimate the sample covariance matrix of the dataset as a whole (i.e., considering together spectral vectors of the two classes)
- Perform the eigenvector decomposition of the sample covariance matrix. You can use Matlab's `eig.m` function, which outputs the matrix containing the eigenvectors as columns, and a diagonal matric containing the eigenvalues on the main diagonal.
    - Note: in the output matrix, eigenvectors/eigenvalues are ordered in **ascending order** of eigenvalue magnitude – therefore, you will need to select columns from the last one and then backwards.
- Choose a number of dimensions K<=220.
- Construct the eigenvector matrix W for K components (i.e., select the last K columns)
- Using W, compute the PCA coefficients for each spectral vector in the test set
- Then from the PCA coefficients obtain an approximation of the corresponding test vector and compute the error (mean square error - MSE)
- Plot the average MSE over the test set as a function of K.
- Plot the eigenvectors corresponding to the 3 largest eigenvalues – this will give you an idea of the basis functions employed by PCA.

Sample code for extracting spectral vectors of class 2:

```
class2=zeros(1428,220);
n=0;
for i=1:size(indian_pines,1)
    for j=1:size(indian_pines,2)
        if indian_pines_gt(i,j)== 2 % class index
            n=n+1;
            class2(n,:)= indian_pines(i,j,:);
        end
    end
end
```

Exercise 2 – Classification using dimensionality reduction and whitening (optional)

In this exercise you will apply a simple 2-class linear discriminant analysis (LDA) classifier to the data belonging to the two classes, before and after **whitening** (i.e. applying PCA plus rescaling

each coefficient to unit variance: $\boldsymbol{y} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{W}^T\boldsymbol{x}$**.** Note that matrix $\boldsymbol{\Lambda}$ is obtained as one of the outputs of `eig.m`). This classifier makes the Naïve Bayes Classifier assumption that the features are statistically independent, thus the shared covariance matrix is taken as $\boldsymbol{\Sigma} = \boldsymbol{I}$. We also assume that class 0 and class 1 are equiprobable. Thus, letting $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ be the mean of vectors in class 0 and 1, we define $\boldsymbol{x}_0 = 0.5(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)$ and $\boldsymbol{w} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$
. A test vector $\boldsymbol{x}$ is classified into class 1 or 0 depending on whether $\text{sign}(\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{x}_0)$ is equal to +1 or -1.
When the classifier is applied to the PCA coefficients, you simply replace $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ with their reduced versions (subtracting $\boldsymbol{\mu}$ and applying the same PCA matrix computed over the training set), and recalculate $\boldsymbol{x}_0$ and $\boldsymbol{w}$ accordingly.

Task: Divide the Indian Pines dataset into training and test data (e.g. 75% of the data of each class to be used as training data, and 25% as test data). Train this classifier on the training data, and apply it to the test data. In particular, you should perform the following:

- Plot the mean vector of class 0 and 1 – this will give you a visual description of the differences among vectors of either class.
- Apply the classifier to the original data (without PCA) and compute its accuracy
- Apply the classifier to the PCA coefficients for different values of K, and compute its accuracy
- Apply the classifier to the original data where only the first K features have been retained, and compute its accuracy (this is a more brutal way to reduce dimensionality)