Assignment 8

Machine Learning: Algorithms and Theory

Prof. Ulrike von Luxburg / Diego Fioravanti / Moritz Haas
Tobias Frangen / Siavash Haghiri

Summer term 2018 — due to **June 19**

1. Handing in the jupiter notebook is mandatory! Please do not write down a report with just results and/or figures. Ideally you should email it, it is easier to correct and more eco friendly, but we accept printed versions too. From now on not handing in the notebook will result in 0 points for the programming part.

2. Before the end of the course you need to present at least one of your solution in the tutorial. If you do not do that you cannot take the exam! If for any reason you cannot attend let us know, it is possible to change group or find another solution.

3. Join the class on ILIAS otherwise we cannot contact you if we need to.

Exercise 1 (Implement PCA, 1+4+2+1+1points) In this exercise you will implement Principal Component Analysis (PCA) and you will see an example of what could go wrong with it.

(a) In `Data` and `Labels` you will find a dataset with two classes. Plot it with two different colours.

(b) Implement PCA as a function `PCA(xs, l)`. It takes as input the data points X_i for $i = 1, \dots, n$ with dimensionality d and l the number of dimensions you want to project on. It returns four values.

   (a) The y_i corresponding to the projection onto $\mathbb{R}^l$. This corresponds to view 2 from the lecture slides.

   (b) The z_i corresponding to the projection embedded in $\mathbb{R}^d$. This corresponds to view 1 from the slides.

   (c) All eigenvalues in descending order.

   (d) The l largest eigenvectors.

   Apply your PCA to the values in `Data` with $l = 1$ and plot the resulting z_i.

(c) Plot the original data with marker `o` and the z_i with marker `x` on the same figure. For every couple (x_i, z_i) plot the line connecting the original data point to its projection.

(d) In `data` you find another dataset. Plot the dataset and repeat (b) and (c) using this dataset.

(e) Here we show what can go wrong with PCA if the dataset is not scaled. In `data` you find a dataset where one component is way larger than the other. Plot the dataset and repeat (b) and (c) using this dataset. What do you notice? Is the projection orthogonal or not?

Exercise 2 (Implement kernel PCA, 1+4+1+2 points) In this exercise you will implement a Kernel Principal Component Analysis (kPCA)

(a) Plot the dataset you find in `data`. Note that you have 3 labels this time.

(b) Implement kernel PCA as a function `kernel_PCA(K, l=2)`. It takes as input the kernel matrix K with is a $n \times n$ matrix and l the number of dimensions you want to project on. It returns the kernel PCA computed with the algorithm from the lecture.

   Note two things: a) due to rounding errors it can happen be that `numpy.linalg.eig` returns complex eigenvectors and eigenvalues. If that is the case a solution is to do `eigenvalues, eigenvectors = eigenvalues.real, eigenvectors.real` b) the matrix $\mathbb{1}_n$ is the matrix $n \times n$ with all entry equal to $1/n$.

(c) Use `sklearn.gaussian_process.kernels.RBF` and compute K for `data` using a RBF kernel with $\sigma = 5$.

(d) Compute the kernel PCA for K with $l = 3$. Plot the resulting points on a 3D figure.

Exercise 3 (Exactly recovery from a PCA is impossible, 2+1 points)

Let x be a vector in $\mathbb{R}^n$. A matrix $W \in \mathbb{R}^{d\times n}$, with $d < n$, induces a map $x \to Wx \in \mathbb{R}^d$. Such map can be interpreted as a lower dimension representation of x. We will call such matrix a *compression matrix*. Similarly if $y \in \mathbb{R}^d$ a matrix $U \in \mathbb{R}^{n\times d}$ induces a map $y \to Uy \in \mathbb{R}^n$. Such map can be interpreted as the reconstruction of y in $\mathbb{R}^n$. Clearly PCA provides one possible way to define such W and U.

(a) Let $W \in \mathbb{R}^{d\times n}$ be an arbitry compression matrix. Show that there exist $u, v \in \mathbb{R}^n$ such that $u \neq v$ and $Wu = Wv$.

(b) Conclude that exactly recovery from a linear compression scheme is impossible.