Assignment 7

Statistical Machine Learning

Sebastian Bordt / David Künstle / Martina Contisciani / Nicolò Ruggeri / Rabanus Derr / Prof. Ulrike von Luxburg

Summer term 2020 — due on **June 18th at 14:00**

Exercise 1 (PCA and SVD, 2 points)

Explain how the Principal Components can be obtained from a SVD of the matrix X.

Exercise 2 (Exact 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 $x \mapsto Wx \in \mathbb{R}^d$. 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 $y \mapsto Uy \in \mathbb{R}^n$. 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 arbitrary compression matrix. Show that there exist $u, v \in \mathbb{R}^n$ such that $u \neq v$ and $Wu = Wv$.

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

Exercise 3 (Feature space, positive semi-definite matrices, 2+2 points)

(a) Find a feature space and a feature map such that the data set represented in Figure 1 becomes linearly separable in the feature space. The feature space representation should be a function of the coordinates (x, y).

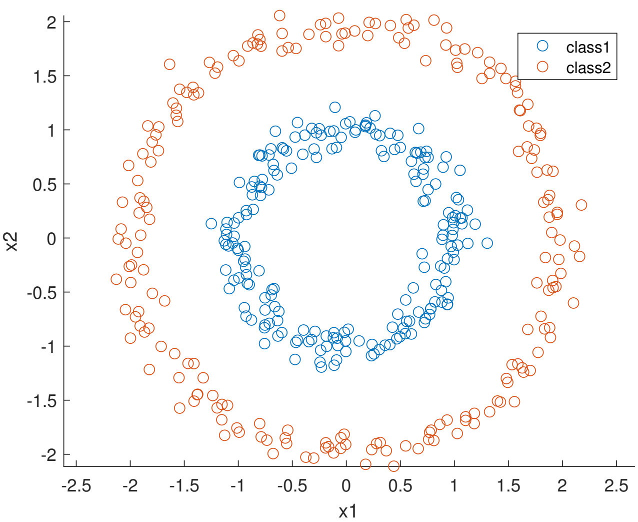


Figure 1: The two classes in this dataset can not be linearly separated in $\mathbb{R}^2$.

(b) Consider a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as $k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathbb{R}^d}$, where $\Phi : \mathcal{X} \to \mathbb{R}^d$ is an arbitrary function. Given a finite set of points $x_1, x_2, \ldots, x_n \in \mathcal{X}$ we define the *Gram matrix* $K \in \mathbb{R}^{n \times n}$ as $K_{ij} = k(x_i, x_j)$. Prove that K is a symmetric and positive semi-definite matrix.

Exercise 4 (Reproducing Kernel Hilbert Space, 1+2 points)

In this exercise we will try to get a better understanding of Reproducing Kernel Hilbert Spaces (RKHS) by looking at the *polynomial kernel*. The polynomial kernel $k_P : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is defined as $k_P(x, y) = (\langle x, y \rangle_{\mathbb{R}^2} + 1)^2$ for any $x, y \in \mathbb{R}^2$.

(a) Show that k_P is a kernel on $\mathbb{R}^2$.

(b) Show that the associated RKHS consists of all functions of the form

$$f(x) = \alpha_1 x_1^2 + \alpha_2 x_2^2 + \alpha_3 x_1 x_2 + \alpha_4 x_1 + \alpha_5 x_2 + \alpha_6,$$

with $x \in \mathbb{R}^2$ and $\alpha \in \mathbb{R}^6$. We denote this space of functions by $\mathcal{H}$. Notice that $\mathcal{H}$ is finite-dimensional. What are the inner product and norm on $\mathcal{H}$? *Hint:* Use a feature map.

Exercise 5 (Design your own exam questions, 3 points) In this exercise, everybody is supposed to come up with suggestions for three exam questions. This is a good way to recap/understand the concepts discussed so far.

Put yourself in our place! We do not want to ask stupid questions. We would like to ask "nice questions". In general, written exams contain three types of questions:

- Questions that are just about **reproducing** knowledge. These kind of questions are pointless and we don't ask questions of this kind in the exam.
- Questions for testing whether the person **understands** the concepts and can apply them to simple situations.
- Questions that require to **transfer** knowledge to new situations.

Your task is now to design exam questions along with their solutions of the two last-mentioned types. The questions can be about any topic covered in the first 7 weeks of the semester (up to video 28). Enter your questions in the LaTex file `my_exam_questions.tex` that we provided and send it to your tutors.

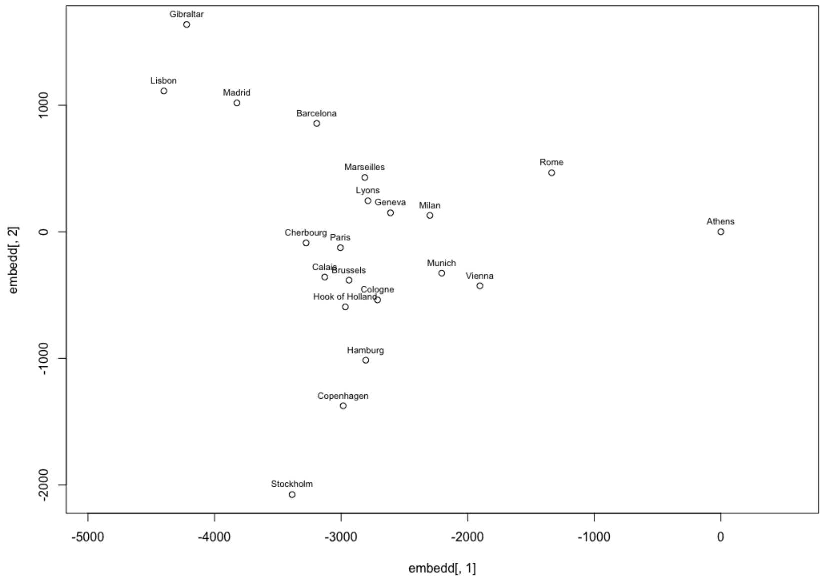
After the class we will put all your questions online. At the end of the course, these questions can help everybody to prepare for the exam!

Exercise 6 (Implement classical MDS, 3+1+1=5 points)

(a) Define a function `MDS` which takes a symmetric $n \times n$ distance matrix `D` and the number of dimensions to embed into `d` as input and returns the embedding `X_embedded`, which is a $n \times d$ dimensional matrix. Try to avoid loops in your implementation and use matrix operations instead.

(b) Load the symmetric $n \times n$ distance matrix from `eurocity.csv`. This gives you the road distances between 21 European cities, where the matrix entry in the i-th row and j-th column corresponds to the distance between city i and j, i.e. the distance between $i = 1$, Athens, and $j = 2$, Barcelona, is 3313km along the shortest road. Apply `MDS` with $d = 2$ to embed the cities into $\mathbb{R}^2$ and plot it. For every point add the names of the cities.

Hint: Your result should look similar to the following figure. You can use the function `plt.annotate` for adding the names of the towns to your plot:



(c) In this task we come back to the USPS data set. In `X` and `y` you can find some of the USPS dataset digits already loaded. They correspond to the test image for the digits $0, 3, 6, 9$. For this subsample of the USPS data set perform MDS with $d = 2$. You have to compute the distance matrix first. For this you can use `euclidean_distances`. Make a scatter plot to see how the digits distribute in the embedding in $\mathbb{R}^2$. Use different colors to plot different digits.