Exercise set #7

Please submit your solutions in teams of two using the sciebo file-drop folder. The link is available in ILIAS. For the formatting please stick to the `submission_guideline.pdf` that you can find on sciebo. In the case of multiple uploads we will consider the latest. Uploads after the deadline will be deleted without further notice.

1. **Eigenvectors and eigenvalues**

   (a) Given a matrix $A \in \mathbb{R}^{n \times n}$ and one of its eigenvectors v, how do you obtain the corresponding eigenvalue λ?

   (b) Show that scaling an eigenvector v of matrix $A \in \mathbb{R}^{n \times n}$ by a constant $c \neq 0$ yields another eigenvector with the same eigenvalue λ.

   (c) For symmetric $A \in \mathbb{R}^{n \times n}$ with distinct eigenvalues $\lambda_1, \ldots, \lambda_n$ show that the corresponding eigenvectors $v_1, \ldots, v_n$ are orthogonal to each other.

   15 points

2. **Centering matrix**
   The Centering matrix is defined as $H = I_n - \frac{1}{n} 1_n 1_n^\top$ where I_n is the $n \times n$ identity matrix, 1_n is the n dimensional one-vector, i.e. $1_n = [1, 1, \ldots, 1]^\top$. Show:

   (a) H is symmetric, i.e. $H = H^\top$.

   (b) H is idempotent, i.e. $H = HH$.

   (c) Show that $H 1_n = 0_n$, i.e., 1_n is an eigenvector with eigenvalue 0.

   (d) What are the other eigenvalues of H?

   (e) For a data matrix $X \in \mathbb{R}^{d \times n}$ show that XH has mean zero, i.e. show $\frac{1}{n} X H 1_n = 0_d$.

   20 points

3. **Benchmarking the centering matrix (programming task)**
   The centering matrix is a very handy tool for theoretical derivations, however it is not a computational efficient way to center a data matrix as you will see by solving this exercise.

   (a) Write a Python function `center_with_matrix(data)` that subtracts the row-wise mean from the input array `data` by multiplying with the centering matrix.

   (b) Write a Python function `centering_with_numpy(data)` that performs the same operation using basic NumPy-functions.

   (c) Sample random data matrices with uniformly distributed entries with 10 rows and a different number of columns. Plot the number of columns of the data matrix against the elapsed runtime for both functions. Also add a legend to your plot.

   15 points

4. **PCA on iris data (programming task)**
For this exercise we will use the iris data set which is available online[1]. The goal is to perform a principal component analysis (PCA) and derive with a new feature subspace. You may use NumPy/SciPy functions for the computations.

   (a) Write a Python function that calculates the covariance matrix
$$\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$
of a dataset $X = [x_1, \ldots, x_n] \in \mathbb{R}^{D \times n}$, where $\hat{\mu} \in \mathbb{R}^D$ is the mean of the data matrix. Do not use the function `np.cov`.

   (b) Implement a function `pca(X, d, whitening=False)` that performs PCA on the input data `X` and returns the projected (and optionally whitened) data `Y`, the matrix of eigenvectors `V`, and the eigenvalues `Lambda`. For the eigenvector decomposition you can use the function `np.linalg.eigh`.

   (c) Project the Iris data onto a two-dimensional feature space. Create scatter plots visualizing the projected data points before and after applying whitening.

Important: This exercise assumes that the data points are stacked columnwise!

30 points

5. **Inhomogeneous kernel function**
For $a, b \in \mathbb{R}^2$, consider the inhomogeneous polynomial kernel function $k(a, b) = (a^\top b + 1)^p$.

   (a) Show that $\Phi(a) = [a_1^2, a_2^2, \sqrt{2}a_1 a_2, \sqrt{2}a_1, \sqrt{2}a_2, 1]^T$ is a feature map for the kernel $k(a, b) = (a^\top b + 1)^2$, i. e., show that $k(a, b) = \Phi(a)^T \Phi(b)$.

   (b) What is the corresponding feature map for $k(a, b) = (a^\top b + 1)^3$? What is the dimensionality of that feature space?

   (c) What is the feature space dimension of $k(a, b) = (a^\top b + 1)^p$? Proof your answer.

20 points

[1] `https://archive.ics.uci.edu/ml/datasets/iris`