Machine Learning and Physics
Winter Semester 2022/2023

Exercise Sheet #1

Due: 31.10.22 11h00

Roman Remme

General Regulations.

- Please hand in your solutions in groups of three people. A mix of attendees from Monday and Tuesday tutorials is fine.
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at https://github.com/hci-unihd/mlph_sheet01. Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter. Please hand in both the notebook (`.ipynb`), as well as an exported pdf.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of three.

1 Principal Component Analysis

Implement PCA from scratch, using only low-level libraries. Assume we have a data set consisting of N observations with p features, summarized in a data matrix $\mathbf{X} \in \mathbb{R}^{p \times N}$. The r first principal components correspond to the eigenvectors of the r largest eigenvalues of matrix $\mathbf{X}\mathbf{X}^T$. After implementing the method yourself, you will apply it to a realistic dataset consisting of simulated hadronic jets, as they are observed by the LHCb experiment at CERN.

(a) Implement PCA in python using `numpy`. Your final implementation should use the vectorized numpy functions or broadcasting instead of loops. Do not assume that the input data is already centered. Hint: `numpy.linalg.eig` (7 pts)

(b) Load the data using the code provided in the jupyter notebook. It consists of simulated measurements of dijets (sets of two jets, i.e. narrow cones of hadrons and other particles produced by the hadronization of a quark). Each sample belongs to one of three classes, originating either from a pair of light quarks (q), charm quarks (c) or bottom quarks (b). How many samples of each class are present in the dataset? What is the range of the different features in the dataset? Normalize them such that each feature has zero mean and unit variance over the samples. Hint: `numpy.mean`, `numpy.std` (4 pts)

(c) Use PCA to reduce the dimensionality of the data to two, such that every sample can be visualized as a point in a 2D scatter plot. Interpret the results; without coloring the points in the plot by class, can you discern distinct clusters? Now use the provided method to color the points by their respective class. How well are the classes separated in the visualization? (4 pts)

2 Nonlinear Dimension Reduction

Apply UMAP to the dataset from Ex. 1.

(a) Use UMAP from umap-learn (https://umap-learn.readthedocs.io/) to reduce the dimensionality of the data to two. As in 1(c), create a scatter plot of the result. Describe and interpret what you observe, only then create another scatter plot, coloring the dots by their ground truth label. How well are the classes separated in the visualization? (2 pts)

(b) Apply UMAP with different values for the `num_neighbors` parameter, ranging from 2 to 100. What do you observe? (1 pt)

(c) For very high dimensional data, it is common to first apply PCA, then UMAP to reduce the dimensionality for visualization. Using either your own implementation from Ex. 1, or PCA from `scikit-learn`, implement this two-stage approach and evaluate it for a varying number of principal components. (2 pts)

3 Bonus: PCA meets Random Matrix Theory

When multiple dimensions are needed to faithfully approximate a distribution, this can be due to either a large intrinsic dimensionality of the data; and / or due to noise.

Here, we explore what happens when we apply PCA to a dataset consisting of pure Gaussian noise. More precisely, let $\mathbf{X} \in \mathbb{R}^{p\times N}$ be a random matrix, with i.i.d. entries from an isotropic Gaussian distribution of full dimensionality, $\mathbf{X}_{ij} \sim \mathcal{N}(0,1)$. We consider dimension reduction with PCA to a subspace of dimension $d \leq p$.

(a) Different draws $\mathbf{X}^1, \mathbf{X}^2, \ldots$ lead to different principal components. What is the distribution of the first principal component? Is it any different for the other principal components? (2 pts)

(b) Intuitively, how do you expect the ordered eigenvalues of $\mathbf{X}\mathbf{X}^T$ to behave, as $p, N \to \infty$ with $p/N \to \lambda \in (0,\infty)$? (1 pt)

(c) Look up the Marchenko–Pastur distribution and use it to deduce a quantitative answer to (b). (2 pts)

4 Bonus: Laplacian Embeddings as Force Fields

Given a graph $\mathcal{G}$ with edges $\mathcal{E}$, the Laplacian Embedding [Belkin 2002] of dimension 1 is given by $z^* \in \mathbb{R}^N$ such that

$$z^* = \underset{\substack{z^T z=1 \\ z^T \mathbf{1}=0}}{\arg\min} \sum_{(i,j)\in\mathcal{E}} (z_i - z_j)^2 \,. \tag{1}$$

Show how z^* can be obtained as one of the solutions to an unconstrained optimization problem corresponding to a force field with attractive and repulsive interactions. Does this generalize this to higher embedding dimension? (5 pts)