

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial).
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using L^AT_EX. For scanned handwritten notes please make sure that they are legible and not too blurry.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at https://github.com/sciai-lab/mlph_w24. Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter. Please hand in your notebook (.ipynb), as well as an exported pdf-version of it.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of two. Specify all names of your group in the submission.

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.

- 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)
- 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)
- 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 labels 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.

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

3 RANSAC

Consider a dataset consisting of a fraction of $p \in (0, 1)$ points sampled from the inlier distribution, while the remaining points are outliers. You are interested in randomly sampling a subset of m points that does not contain any outliers. How many repeats r are needed to find at least one outlier-free subset with a probability of 99%? You can assume that each subset is sampled with replacement, i.e. it may contain duplicates.

Hint: For a single m -subset, what is the probability of it being outlier-free? (3 pts)

4 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 $\mathbf{X}^1, \mathbf{X}^2, \dots$ lead to different principal components. What is the (directional) 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 $p, N \rightarrow \infty$ with $p/N \rightarrow \lambda \in (0, \infty)$? (1 pt)
- (c) Look up the Marchenko–Pastur distribution and use it to deduce a quantitative answer to (b). (2 pts)