Exercise Sheet 1

Due: 27.10.25 14h00

Fred Hamprecht, Tristan Bereau
Gerrit Gerhartz, Christof Gehrig

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial group).
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. For scanned handwritten notes, please ensure they are legible and not blurry.
- For the practical exercises, 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`) and an exported PDF.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of two. Specify all names of your group in the submission.
- You can find all the data in the GitHub Repository.

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` (5 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` (3 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 labels to color the points by their respective class. How well are the classes separated in the visualization? (3 pts)

2 Robust PCA

Now look at the artificial two-dimensional data that contains an outlier. You don't need to center data in this exercise.

(a) Perform standard PCA and plot the first principal component in a scatter plot of the data. (1 pt)

(b) Compute the first principal component in a robust way: Use the Tukey potential with a scale parameter of $s = 1.5$ on the distances,

$$T(r;s) = \begin{cases} \frac{s^2}{6}\left[1 - \left(1 - \left(\frac{r}{s}\right)^2\right)^3\right] & \text{for } |r| \leq s \\ \frac{s^2}{6} & \text{for } |r| > s \end{cases}.$$

Parameterize the line with the angle φ to the x-axis. Plot the error as a function of φ. Interpret the two minima. (3 pts)

(c) Plot the potential in the scatter plot for the two minima φ^*. (2 pts)

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 want to fit a good initialization for a robust estimator. You repeatedly sample a subset of m points, in the hope of obtaining at least one subset which is free from contamination. 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 $\mathbf{X}^1, \mathbf{X}^2, \ldots$ 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 $p, N \to \infty$ 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)

~~③ Define:~~

- ~~N_{in} : # inliers~~
- ~~N_{out} : # outliers~~
- ~~N_m : # points in subset~~

~~Case for $r = 1$. Picking m times without putting back.~~

~~Use hypergeometric distribution, where X: No outliers are present in m.~~

$$P(X) = \frac{\binom{k}{m}\binom{N-k}{n-m}}{\binom{N}{n}} \overset{m=0}{=} \frac{\binom{N_{in}+N_{out}-N_{out}/N_{in}}{N_m}}{\binom{N_{in}+N_{out}}{N_m}}$$

~~and with~~ $\binom{N}{k} = \frac{N!}{k!(N-k)!}$:

$$P(X) = \frac{(N_{in}+N_{out}-N_{out}/N_{in})!}{\cancel{N_m!}\,(N_{in}+N_{out}-N_{out}/N_{in}-N_m)!}\left(\frac{(N_{in}+N_{out})!}{\cancel{N_m!}\,(N_{in}+N_{out}-N_m)!}\right)^{-1}$$

$$= \frac{(N_{in}+N_{out}-N_{out}/N_{in})!\,(N_{in}+N_{out}-N_m)!}{(N_{in}+N_{out}-N_{out}/N_{in}-N_m)!\,(N_{in}+N_{out})!}$$

Did not see the "with replacement"

For $r = 1$ independent pick:

$\Rightarrow P(\text{subset is outlier-free}) = p^m$

$\Rightarrow P(\text{subset contains} \geq 1 \text{ outlier}) = 1 - p^m =: P(X)$

For r independent picks:

$\Rightarrow P(X) = (1-p^m)^r$

$\Rightarrow P(\text{at least one subset is outlier-free}) = 1-(1-p^m)^r =: P(Y) \overset{!}{\geq} 0.99$

$\Rightarrow (1-p^m)^r \leq 0.01$ | ⚠ Multiplied with negative number, as $0 \leq (1-p^m)^r \leq 1$

$\Rightarrow r \geq \frac{\ln(0.01)}{\ln(1-p^m)}$