Assignment 8

Statistical Machine Learning

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

Summer term 2020 — due on **June 25th at 14:00**

Exercise 1 (k-means and kernel k-means, 2 + 2 points)

(a) Show that the cluster centers are the best representatives of a cluster in the sense that for any cluster, the cluster center m satisfies equation (1). (See the slides for understanding notation).

$$\sum_{i \in C_k} \|X_i - m_k\|^2 \leq \sum_{i \in C_k} \|X_i - X_j\|^2, \forall j \in C_k \tag{1}$$

(b) Prove that on any given data set of n points in $\mathbb{R}^d$, the k-means algorithm terminates after a finite number of steps (hint: as an intermediate step, show that in each iteration of the while loop, the objective function cannot increase; you may use the result from part (a).)

Exercise 2 (Implement Isomap, 4+2 points)

(a) Define a function `Isomap`. It takes a symmetric $n \times n$ distance matrix `D`, a parameter `k` for the KNN-graph and the number of dimensions `d` as input and returns the Isomap embedding `X_embedded` into $\mathbb{R}^d$.

Hint 1: The NetworkX tutorial is a good place to start for Python and graphs. Remember that you want the path length and not only the path.

Hint 2: When creating a network for Isomap, edges are weighted using the euclidean distance between points. Make sure that this is the case in your implementation. Notice that Isomap is one of the few algorithms that uses the distances for edge weights. Most algorithms weight edges by similarity rather than distance.

(b) Apply Isomap to the provided USPS dataset with $d = 2$. Vary $k \in \{5, 20, 50\}$ and plot the resulting embeddings. A good implementation of Isomap should not take longer than a minute for this computation. Which embedding seems to be the most reasonable, and what happens for $d = 1$?

Exercise 3 (PCA and Random Projections, 1+1+3+1+1+2+1=10 points + 3 Bonus points)

In this exercise we explore Principal Components Analysis and Random Projections on the USPS dataset. For this, you can either use `PCA` and `GaussianRandomProjections` as provided by `scikit-learn`, or you can implement these methods yourself. If you choose to implement these methods yourself it will be awarded with 1.5 bonus points per methods.

(a) Use PCA and Random Projections to reduce the dataset's dimensionality to $d = 2$. Visualize the projected data. Use different colors for different digits.

Hint: Random projections are *random*, and we do not use seeds in this exercise.

(b) Visualize the principal components and random directions as images in feature space.

We want to compare the reconstruction error of PCA and Random Projections. To compute the reconstruction error, we have to transform the projected data back into the original space. `PCA`

offers a method `inverse_transform` that performs this task. In case of Random Projections, we are looking for the matrix $U \in \mathbb{R}^{256 \times d}$ that minimizes

$$\min_{U \in \mathbb{R}^{256 \times d}} \sum_{i=1}^{n} ||UWx_i - x_i||_2^2.$$

Here $W \in \mathbb{R}^{d \times 256}$ is the random projection matrix and Wx_i is the projection of point x_i.

(c) Implement a function `reconstruct` that computes the matrix U and transforms the randomly projected points back into the original space.

(d) For different values of d, visualize the reconstructed images in the feature space.

(e) Compare the reconstruction error of PCA and Random Projections. The reconstruction error is the euclidean distance between an original and a reconstructed point. Plot the average reconstruction error for different dimensions d.

The reconstruction error is not the only measure that aims to describe the quality of an embedding. There are also measures of distance distortion. We now want to compare the average and maximum distance distortion of PCA and Random Projections. If you are interested, you can read about different measures of distortion in Vankadara & von Luxburg (2018).[1]

(f) The distance distortion for a pair of points (x, y) is the ratio of point distances between the original points (x, y) and the projected points $(\tilde{x}, \tilde{y})$.

$$\rho = \max \left\{ \frac{d(x,y)}{d(\tilde{x},\tilde{y})}, \frac{d(\tilde{x},\tilde{y})}{d(x,y)} \right\}$$

Implement a function `pairwise_distortion` that computes the distance distortion for all pairs of points. Compute and plot the average and maximum distance distortion for PCA and Random Projections for different dimensions d.

(g) Summarize the advantages and disadvantages of PCA and Random Projections. Draw on your results in this exercise and what you learned in the lecture.

Hint: This exercise relates to the Johnson-Lindenstrauss Lemma. For Random Projections, especially if you want to implement them yourself, refer to the book by Shai Shalev-Shwartz and Shai Ben-David.

[1] `http://www.tml.cs.uni-tuebingen.de/team/leena_chennuru/VankadaraLuxburg18.pdf`