Foundations of Machine Learning
Practical Course | SS 2024 – 3
Prof. Michael Botsch, Marion Neumeier

Exercise Sheet 3

1. Exercise: Dimensionality Reduction

(a) What categories of dimensionality reduction do you know? Name one example technique for each of the categories.

(b) Define the term "curse of dimensionality". How does dimensionality reduction help to overcome this problem?

(c) What are some of the potential limitations or drawbacks of t-SNE? What are some of the potential limitations or drawbacks of PCA?

2. Exercise: Principal Component Analysis (PCA)

(a) Load the digits dataset provided by the scikit-learn (`sklearn`) library and plot one exemplary image for each digit 0-9 as follows using the `matplotlib` library. Use one figure with multiple axes to plot all digits.

(b) How many samples M does the dataset contain? And how many features N does each sample contain? Print this information in your code.

(c) Perform the Principal Component Analysis (PCA) on the data $\mathbf{V} \in \mathbb{R}^{M \times N}$ step by step without utilizing any built-in functions provided by any libraries, unless specified otherwise.

(i) Calculate the features expected value (mean) $\hat{\mu}_n$ from the data points $\boldsymbol{v}_m$ and use it for centering of features by applying

$$v_{m,n} = v_{m,n} - \hat{\mu}_n, \qquad \text{with} \qquad \hat{\mu}_n = \frac{1}{M} \sum_{M}^{m=1} v_{m,n}. \tag{1}$$

(ii) Calculate the sample covariance matrix $\hat{\mathbf{C}}_v$ from the **centered** data points.

(iii) Perform eigendecomposition to determine the eigenvalues $\lambda_i = \Lambda_{ii}$, where $\mathbf{\Lambda} \in \mathbb{R}^{N \times N}$, and eigenvectors $\mathbf{U} \in \mathbb{R}^{N \times N}$ by solving

$$\hat{\mathbf{C}}_v = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}. \tag{2}$$

Tip: Use the function `np.linalg.eigh()` (`numpy` library) to get the eigenvalues and eigenvectors.

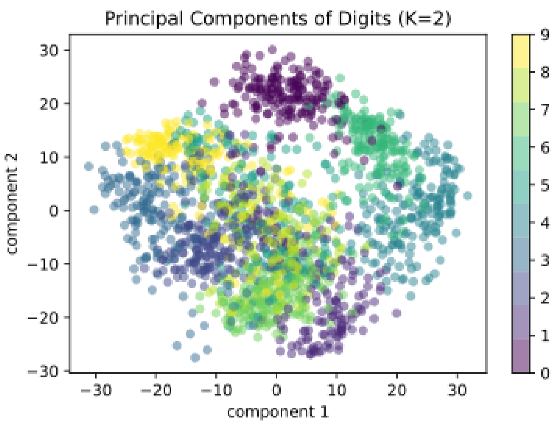
(iv) What do the rows of $\mathbf{U} \in \mathbb{R}^{N \times N}$ denote? What do the columns of $\mathbf{U} \in \mathbb{R}^{N \times N}$ denote?

(v) Select the $K = 2$ eigenvectors corresponding the biggest K eigenvalues such that $\mathbf{U}_K \in \mathbb{R}^{N \times K}$.

(vi) Perform the dimensionality reduction by computing

$$\mathbf{V}_{\text{proj}} = \mathbf{U}_K^{\mathrm{T}}\mathbf{V}. \tag{3}$$

(vii) Plot the projected parameter space $\mathbf{V}_{\text{proj}}$ with the according label information using `matplotlib` library. The resulting image should look similar to this:



Tip: If your plot looks stretched or mirrored your result is not necessarily wrong. In some cases, a matrix may have multiple valid decompositions with different eigenvectors and eigenvalues.

3. Exercise: t-SNE

Carry out the initial steps of t-SNE on the data matrix $\mathbf{V} \in \mathbb{R}^{M \times N}$ without utilizing any built-in functions provided by any libraries, unless specified otherwise. The steps are as follows.

(a) Implement function `compute_p_j_given_m()` that computes and returns the conditional probability $p_{j|m}$ for a data sample $\boldsymbol{v}_m \in \mathbb{R}^N$, where $m = 1, \ldots, M$. $p_{j|m}$ indicates the probability that $\boldsymbol{v}_m$ would select point $\boldsymbol{v}_j \in \mathbb{R}^N$ as its neighbor if the neighborhood is selected proportional to its probability density under a Gaussian PDF centered around $\boldsymbol{v}_m$.

$$p_{j|m} = \frac{e^{-\frac{||\boldsymbol{v}_m - \boldsymbol{v}_j||^2}{2\sigma_m^2}}}{\sum_{k \neq m} e^{-\frac{||\boldsymbol{v}_m - \boldsymbol{v}_k||^2}{2\sigma_m^2}}} \tag{4}$$

(b) Implement function `compute_perplexity()` that computes and returns the perplexity $Perp(\mathrm{P}_m)$ for the m-th data sample using

$$Perp(\mathrm{P}_m) = 2^{H(\mathrm{P}_m)} \tag{5}$$

$$\text{where} \quad H(\mathrm{P}_m) = -\sum_{j \neq m} p_{j|m} \log_2(p_{j|m}). \tag{6}$$

(c) Implement function `binary_search()`, performing binary search, which can be applied to optimize the standard deviation σ_m assigned to each data sample $\boldsymbol{v}_m$. The function's default parameters are set as follows: search boundaries $l_{\text{min}} = 0$ and $l_{\text{max}} = 100$, the maximum number of iterations $I = 100$, and a tolerance level of $\epsilon = 0.001$. The search stops when the range of possible solutions is narrowed down to a length less or equal to ϵ, even if the maximal iteration number is not reached.

(d) Load the `tsne_data.pkl` dataset provided in the Moodle course using the `pandas` library, which consists of the datapoints $\mathbf{V} \in \mathbb{R}^{M \times N}$, where $M = 20$ and $N = 2$. Use the implemented functions (a)-(c) to determine the standard deviation σ_m for each data sample $\boldsymbol{v}_m$. The desired perplexity is $Perp = 10$. The search boundaries for the algorithm are set to $l_{\text{min}} = 0.15$ and $l_{\text{max}} = 50$.