Assignment #4

Elements of Machine Learning

Saarland University – Winter Semester 2024/25

Rabin Adhikari
7072310
raad00002@stud.uni-saarland.de

Dhimitrios Duka
7059153
dhdu00001@stud.uni-saarland.de

3 Problem 3 (Dimensionality Reduction)

3.1 What information does the first principal component capture in terms of the data variance and the data explaining?

The first principal component is the direction in which the data varies the most. In other words, it's the direction that captures the most variance in the data. Furthermore, assuming that the underlying data is linearly distributed, the first principal component represents the closest line to the data.

3.2 Calculate the first principal component.

We can imagine the provided data as a matrix where each row represents a data point and each column represents a feature.

$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

First, we have to normalize the data. To perform normalization, first we compute the per feature mean μ_i and per feature standart deviation σ_i.

$$\begin{aligned} \mu_1 &= \frac{1+2+3}{3} = 2 \\ \mu_2 &= \frac{1+2+3}{3} = 2 \\ \sigma_1 &= \sqrt{\frac{1}{2}(1+0+1)} = 1 \\ \sigma_2 &= \sqrt{\frac{1}{2}(1+0+1)} = 1 \end{aligned} \tag{1}$$

Applying the normalization formula, we get the following matrix:

$$X = \begin{pmatrix} -1 & -1 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

Next, we compute the covariance matrix.

$$\begin{aligned}\Sigma &= \frac{1}{n-1}X^T X \\ &= \frac{1}{2}\begin{pmatrix}-1 & -1\\ 0 & 0\\ 1 & 1\end{pmatrix}^T \begin{pmatrix}-1 & -1\\ 0 & 0\\ 1 & 1\end{pmatrix} \\ &= \begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\end{aligned} \tag{2}$$

Next, we have to compute the eigenvectors and eigenvalues of the covariance matrix.

$$det\begin{pmatrix}1-\lambda & 1\\ 1 & 1-\lambda\end{pmatrix} = 0$$

$$(1-\lambda)^2 - 1 = 0 \tag{3}$$

Solving the equation above, we get $\lambda_1 = 0$ and $\lambda_2 = 2$. This means the the eigenvector related to $lambda_1$ doesn't capture any variance. Therefore, we are interested in the eigenvector associated with λ_2.

$$\begin{aligned}(\Sigma - 2I)v &= 0 \\ \begin{pmatrix}-1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}v_1\\ v_2\end{pmatrix} &= 0\end{aligned} \tag{4}$$

Solving the equation above, we get the following eigenvector: $v_1 = \begin{pmatrix}1\\ 1\end{pmatrix}$. Normalizing v_1, we get: $v_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix}$.

3.3 Can PCA be used to reduce the dimensionality of a highly nonlinear dataset? Explain.

PCA can be applied to reduce the dimensionality of a highly nonlinear dataset, but the results are often unsatisfactory. This is because PCA is fundamentally a linear technique. Its goal is to identify linear combinations of features that capture the maximum variance in the data by finding orthogonal directions that represent the directions with the highest variance. However, for nonlinear datasets, PCA struggles to capture the complex relationships between features, leading to the loss of important information and failing to preserve the dataset's underlying structure. A simple example is the spiral dataset. Suppose we have a spiral dataset in 2D and apply PCA to reduce it to 1D. The data would end up being projected into a single line, therefore losing its initial structure.

3.4 When might be sensible to chain two different dimensionality reduction algorithms? You can support your answer with an example.

Applying t-SNE to a large dataset is computationally expensive as it computes pairwise conditional probabilities for each data point. The solution to this problem is to use a combination of t-SNE and PCA. First, we use PCA to reduce the dimensions to a reasonable number of features, and after that, we run t-SNE to further reduce the dimensionality of the data.

3.5 How can you assess the effectiveness of a dimensionality reduction algorithm, used as a preprocessing step, on your dataset by considering the accuracy or error of a downstream model?

We can assess the effectiveness of a dimensionality reduction algorithm, used as a preprocessing step, on a dataset in the following way. First, we train the model using the full, nondimensional-reduced data and evaluate its performance on a downstream task $\mathcal{X}$. Afterward, we apply the dimensionality reduction technique to the data, train the model, and evaluate its performance again. Finally, we compare the two performances with each other.