PCA and dimensionality reduction in ML

Starting with the knowledge gained during webinar classes explain what is PCA and why we would like to use it within a Machine Learning processing pipeline. By using the last example we discussed when talking about the PCA explain what is the meaning of the "dimensionality reduction" and how we could use it in practice. Please put the essay in your confluence space.

PCA technique in Machine Learning

Principal Component Analysis (PCA) is one of the most commonly used unsupervised machine learning algorithms across a variety of applications: exploratory data analysis, dimensionality reduction, information compression, data de-noising, and much more. The main aim of PCA is to find such principal components, which can describe the data points with a set of principal components. Those principal compenents are vectors.

It is a method of feature extraction which groups variables in a way that creates new features and allows features of lesser importance to be dropped.

How to do a PCA?

1. Standardize the range of continuous initial variables
    - The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis. To prevent some problems that may appear with large differences between the ranges of initial variables we transform the data to comparable scale.
2. Compute the covariance matrix to identify correlations
    - The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. It's all due to the fact that variables are highly correlated in such a way that they contain redundant information. So that's why we need a p x p covariance matrix where p is the number of dimensions.
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
    - Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the?principal components?of the data. Choosing this way, we are allowed to reduce the dimensionality without loosing too much information that may be important.
4. Create a feature vector to decide which principal components to keep
    - The aim of this step is to create a simple 1D matrix - a vector hat has as columns the eigenvectors of the components that we decide to keep. So it's our choice wether we want to keep the components or not.
5. Recast the data along the principal components axes
    - The aim of this step is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components. This may be easily done by multiplying the transpose of the original data set by the transpose of the feature vector.

Summary

PCA does have some issues. The major one is that the results are directly dependent on the scale of the variables. In spite of that PCA is a safe method of feature extraction and dimensionality reduction and should be used to understand the relationships between variables in extremely large data sets.

Bibliography:

https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/

https://www.keboola.com/blog/pca-machine-learning