Assignment on PCA & Anomaly Detection

Machine Learning

Prof. Dr. Marc Pouly & Dr. Tim vor der Brück

Lucerne University of Applied Sciences and Arts

Deadline: May 1, 2018.

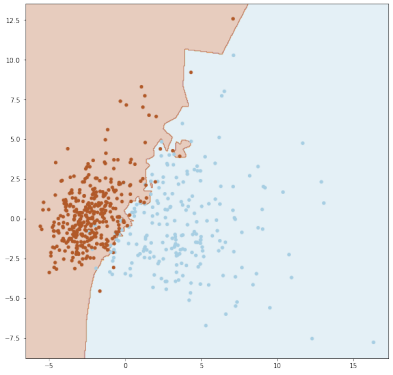
Exercise 1: Theory Questions (1 Point)

Fill in the online theory quiz on ILIAS.

Acceptance criterion: This is an individual exercise. You can participate in the quiz as many times as you want. ILIAS registers your best result over all runs. You passed this exercise if your best score reaches at least 80%.

Exercise 2: Decision Boundary (0 Points)

The decision boundary of a machine learning classifier is a very important concept that visually supports us in understanding what exactly a classifier does and how it performs. We provide a Jupyter notebook that displays the decision boundary of a k-NN classifier applied to the breast cancer dataset. Inspect the code and teach yourself in creating and interpreting decision boundaries for different classifiers. This is a self-study exercise, you do not need to hand in anything.



Exercise 3: Principal Component Analysis (1 Point)

This is a team exercise; hand in your solution as a Jupyter notebook by e-mail to Marc Pouly no later than the above deadline.

3.1 PCA for Data Inspection and Redundancy Analysis

The HSLU campus in Horw operates a parking ground for students, visitors and staff. Its barrier is equipped with a REST interface from which we can query the number of free parking spots.

Download this data set from ILIAS and proceed as follows:

1. Feature engineering: replace the time stamp by individual columns for year, month, day, hour and minute (use `pandas.DatetimeIndex`). Introduce dummy variables for string attributes (use `pandas.get_dummies`) and convince yourself that your data set matches a vector space model.

2. Use Principal Component Analysis to visualize the loss of information (i.e. variance) under dimensionality reduction, i.e. print the number of features on the x-axis and the retained variance on the y-axis. Interpret this visualization in your own words; what do you conclude about this data set; would you suggest data cleaning measures and do you think this data is suitable for training a machine learning regression model that predicts the number of parking spots available.

3. Refactor your code towards a generic redundancy analysis template for arbitrary data sets. Measure redundancy in the tourism data set from assignment 1.

3.2 PCA for Visualization

Take the AutoScout24 dataset reduced to the numeric attributes (1) year, (2) mileage, (3) horsepower, (4) engine size, (5) seats, (6) cylinders and (7) gears. Use PCA to reduce this dataset to only 2 dimension and display the data as a 2D scatterplot. There may be too many data point for visualization such that you may want to reduce the number of records with sampling. Finally, use the price information of each car to colour your scatterplot. What conclusions can you draw from your visualization?

Exercise 4: Anomaly Detection (1 Point)

This is a team exercise; hand in your solution as a Jupyter notebook by e-mail to Marc Pouly no later than the above deadline.

You watched the video tutorial by Andrew Ng on anomaly detection using the multivariate Gaussian distribution. There is a data set on ILIAS containing credit card transactions made available by a European bank in September 2013 on Kaggle. Not surprisingly, the data set is anonymized, i.e. a PCA transformation was executed on all features (i.e. each feature therefore is a linear combination of the original but unknown features). The only exceptions are amount and time (milliseconds since first transaction). Finally, there is a class feature to indicate whether the transaction is fraudulent (value: 1) or normal (value: 0). Implement fraud detection using the statistical approach introduced by Andrew Ng:

1. Remove the time feature. It is ill-designed and just makes your life complicated (by generating a singular covariance matrix)

2. Observe the extreme class imbalance, which obviously is typical for anomaly detection. Create a test set that consists of all fraudulent transactions and, randomly drawn, the same number of normal transactions. Take the remaining normal transactions as training set.

3. Determine mean vector (`mean`) and covariance matrix (`cov`) from the training set and fit a multivariate Gaussian distribution. Instead of programming all the formulas yourself, you may want to use Python's built-in function (`scipy.stats.multivariate_normal`) for fitting a multivariate Gaussian distribution from (sort-of independent) single features:

```
p = multivariate_normal(mean, cov)
densities = p.pdf(test_data)
predictions = (densities < threshold)
accuracy_score(test_data['Class'], predictions)
```

4. Implement a method that helps you choose a good threshold value.