LINMA 2472 ALGORITHMS IN DATA SCIENCES: HOMEWORK II

HW 2 k-PCA and random projections

Please prepare a written report (**max** 5 pages) with appropriate figures based on the results from the assignments. You also need to provide and fill the associated jupyter notebook (see, *e.g.*, `https://jupyter.org/`).

The deadline for this assignment (and the next part to come) is on Sunday 2021-11-14 23:59 Brussels time.

Assignment 1: PCA vs. k-PCA for outliers detection

In this assignment, you will compare PCA and k-PCA for the detection of outlier. We define an outlier as a point that is sampled from another distribution than the one we consider.

For the sake of avoiding the reinvention of the wheel, you can use the PCA and kPCA implementations of `sklearn` (`https://scikit-learn.org/stable/`).

Q1. (EDA) Perform a bit of EDA (exploratory data analysis). Compute the outlier ratio r_o, plot the *true* data and the outliers. Comment on the shape of the *true data* vs. *outlier*: do you believe the task of separating both will be easy?

In this homework, we consider as success metric the number of positive hits, that is

$$\texttt{score} := \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}},$$

with the same notation as in the slides.

What is the minimum score that is acceptable for any outlier detector that we design?

Q2. (Train/Test/Val) In order to have a clean ML workflow, we will separate our dataset into three parts: the train and validation sets, that form the data *available* to find the best model and train it, and the *test* set that will be used **once and only once** at the very end. Such workflow is illustrated in Fig. 1.1.

You can find more information about the train/test split on Wikipedia (`https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets`), note that here we will not go into cross-validation (even if it is generally a good idea to do so).

Using the `train_test_split` function from `sklearn`, split the data into train, test and validation with 10% of test set, and a ratio 70/30 between the train and validation sets.

Do you need to shuffle (`shuffle`=True), or not?

Q3. (PCA) Let us perform outlier detection with `PCA`. We will use the abstract class of `sklearn` to quickly and efficiently build our classifier as a new class(ifier) denoted as `OutlierDetectorPCA`.

We already implemented the constructor (`__init__`), and several class functions. However, there are some holes in the instructions of some of these functions; you are asked to fill it.

Test your implementation by computing the validation score for $k = 1$, is it an acceptable score?

Discuss the value of k. Is there any other hyperparameter to tune during the model selection? Refer to `https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)` for the distinction between parameters and hyperparameters.

What is/are the parameter(s) of the model?

Q4. (k-PCA) Let us now perform the outlier detection with `k-PCA`.

For the sake of simplicity, and for educational purpose we will "cheat", that is *we will explicitly compute the features*.

Using the `kpca` implementation of `sklearn`, we have access to this information. Note that for a *real* application where the number of features is very high, we won't be able to use the features, and we will have to use the k-PCA implementation presented in the slides.

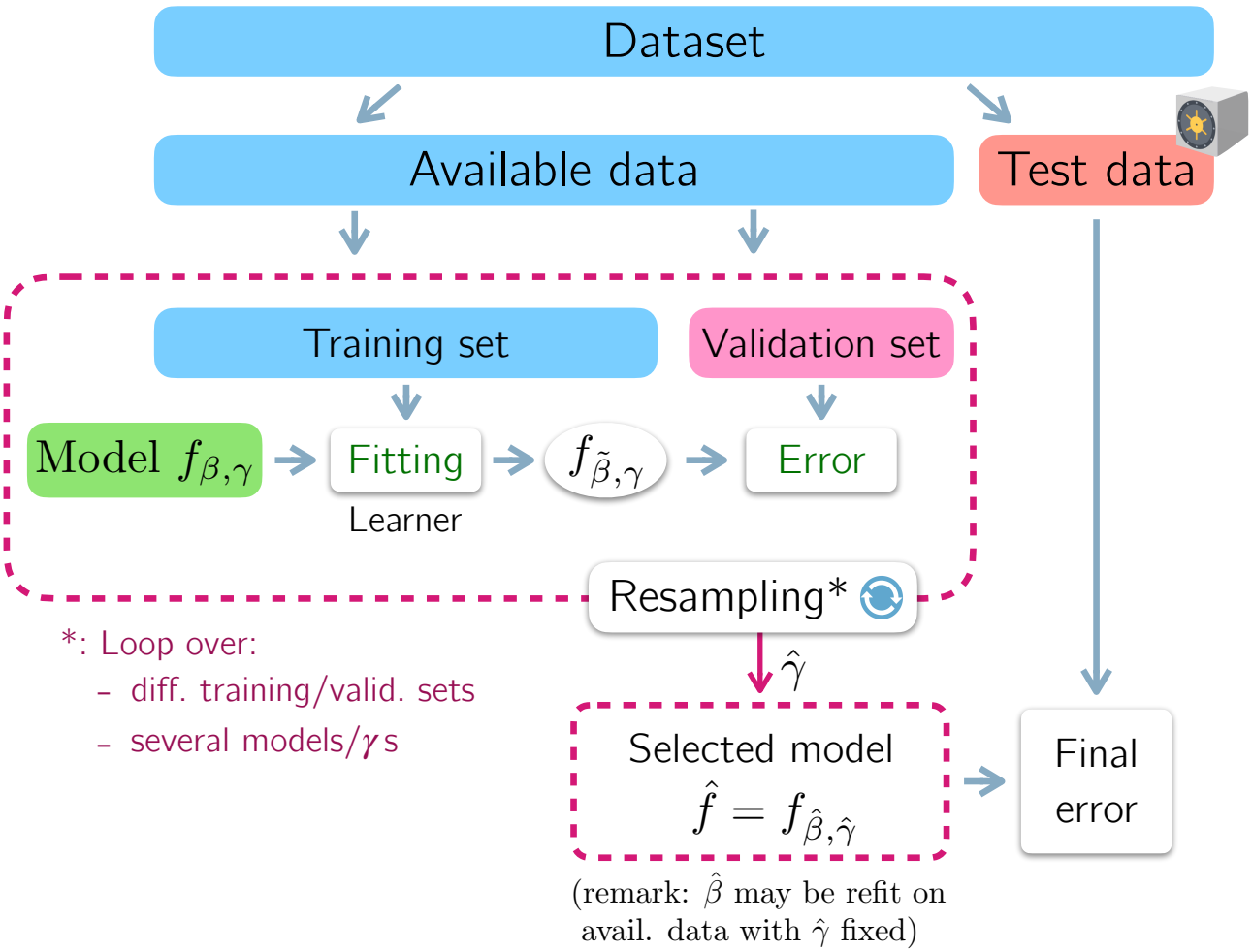

Figure 1.1: Illustration of a workflow for training, tuning and testing models. Image credit: Laurent Jacques for LEPL1109.

Hence, you should *reuse* your answer of Q3, by writing a class `OutlierDetectorKPCA` that *inherits* from `OutlierDetectorPCA`.

Explain shortly (max 8 lines) what is done in the function `fit` from the class `OutlierDetectorKPCA` that we implemented for you.

Implement the function `predict` from the class `OutlierDetectorKPCA`.

Discuss briefly the value of k: in which range does it lie?

What are the hyperparameters of this model?

What is/are the parameter(s) of the model?

Q5. (Model selection) Using the train and validation sets, select the best model, among `PCA` and `kPCA`. For the latter, test any combination of the parameters with $k \in \{1, 5, 10, 50\}$, $\gamma \in \{0.001, 0.01, 1, 10, 100\}$, `kernel` $\in \{$`rbf`, `poly`$\}$.

Give a brief interpretation of each of the parameter, and the optimal combination.

Plot and comment the ROC of an outlier detector using PCA, and the best outlier detector using k-PCA.

Using `plot_comparison_outlier_detector`, discuss whether the detectors do well (or not) their job.

Q6. (Analysis) Evaluate the generalization error on your best classifier with the test set. Comment briefly on what you obtain.

Does your (best) outlier detector detect any potential outlier that may be located far away from the bulk of data?

If not, propose an improvement of the reconstruction error metric to alleviate the problem. *hint: instead of considering the absolute (reconstruction) error, use the relative one.*

Is there any other positive behaviour from this change? *hint: look at the range of the metric.*