INF264 - Exercise 5

Natacha Galmiche

1 Linear SVM

In this first exercise, we get familiar with the linear SVM classifier on simple datasets. A template is here to guide you, notably for question 4.

1. Load and split the dataset contained in the file `svm_data_1.csv` into 3 datasets: a training, validation and test dataset.
2. How many features are they? What type of target does the dataset contain? Visualize the dataset. Is it linearly separable?
3. Train and evaluate the performance of a very simple `sklearn.svm.SVC` model, where the argument `kernel` is set to `linear`. You can set the argument `C` to a very high value like 1E10 (explain why).
4. Visualize the hyperplanes and compute the maximized margin. **Hint:** More details about the definition of these hyperplanes, how to compute them and how to visualize them using sklearn is available in the template!
5. Load and visualize the dataset contained in the file `svm_data_2.csv`. Is it linearly separable? How would you define it then?
6. Which argument of the `sklearn.svm.SVC` model should you tune in order to learn this type of data? What does this hyperparameter control?
7. Train different SVC models with different values for this hyperparameter.
8. Select the best model using a classic model selection pipeline.
9. Visualize the hyperplanes and compute the maximized margin and comment.

2 Bagging

A bootstrap data sample is a sample of a dataset with replacement. Bootstrap aggregating is a method in machine learning that can be used in some of the high-variance machine learning algorithms such as decision tree in order to prevent overfitting.

The dataset we will use in this exercise is the Sonar dataset. This dataset describes sonar chirp returns bouncing off different surfaces. The 60 input variables are the strength of the returns at different angles. It is a binary classification problem with 208 observations that requires a model to differentiate rocks (labeled R) from metal (labeled M) cylinders.

There is a template available, you can choose whether you want to use it or not.

1. Load the sonar dataset and store the features in a matrix X and labels in a vector y. Split the dataset for a cross validation pipeline.
2. Bootstrap sample (slide 14 of lecture 10 on ensemble):
   (a) Write a `bootstrap_sample` function that returns random subsamples with replacement `X_sample` and `y_sample` from a feature dataset `X` and its corresponding target dataset `y`. Note that `X_sample` and `y_sample` may contain duplicates. You can use the function `random.choices()` to generate subsampling with replacement.
   (b) Call this function on the training dataset and compute the ratio
   'unique occurrences of datapoints in `X_sample`' / 'length of `X_sample`'. Compare this ratio to the one mentioned in the lecture slide on 'Bootstrap sampling'.
3. Bagging training function. Implement the Bagging training algorithm as explained in the lecture 10: ensemble, slide 15.

(a) This function should take as input a training dataset `X_train` and `y_train`; a number of classifiers `n_cfls`; a classifier `Classifier` (could be any sklearn class implementing a classifier).

(b) This function should return a list `cfls` of `n_cfls` trained classifiers composing the bagging classifier.

4. Bagging predict function. Implement the Bagging training algorithm as explained in the lecture 10: ensemble, slide 15.

   (a) This function should take as input a list of trained classifiers `cfls` composing a bagging classifier and some feature values `X`.

   (b) This function should return a vector of predictions made by the bagging classifier with the rule of majority vote.

5. Bagging pipeline, using cross validation on the sonar dataset using decision trees:

   (a) Train different boosting classifiers based on different numbers of `n_clfs` classifiers. Select the best model using cross validation.

   (b) Evaluate the best model.

   (c) Compare the performance on the test set of a bagging classifier consisting of a single Decision Tree (that is to say the bagging classifier is actually a decision tree) with your selected model.

   (d) What can we say about the variance and bias of a simple decision tree compared to a bagging classifier?

   (e) What are the disadvantages of a bagging classifier then?