Classification Empirical Study

1 Introduction

In this course, you have seen two approaches to classification. The first is logistic regression, a type of generalized linear model. The second is the neural network. In this project, you have the opportunity to understand two other popular and quite different classification methods: decision trees and K-nearest neighbours (KNN). The main task will be to perform an empirical study of three approaches (logistic regression, decision trees, KNNs) using a variety of metrics and plots.

1.1 Decision Trees

Read Section 14.4 of *Pattern Recognition and Machine Learning* by Bishop (2006) for background on decision trees.

1.2 K-Nearest Neighbours

Read Section 1.4 of *Machine Learning* by Murphy (2012) for background on K-nearest neighbours: `https://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-22may12.pdf`.

2 Tasks

2.1 Task 1: Setting Up

- Load the Iris data set, a famous data set of plant attributes by R. A. Fisher, using `sklearn`:

```
from sklearn.datasets import load_iris
iris = load_iris()
X, Y = iris.data, iris.target
```

- Familiarize yourself with the descriptions of the `KNeighborsClassifier`[1] and `DecisionTreeClassifier`[2] in the `sklearn` documentation.

2.2 Task 2: Fitting the Data

- Use the `fit` method of the `KNeighborsClassifier` and `DecisionTreeClassifier` to fit models to X, Y.
- Use your existing implementations (either from Day 2 or Day 3) to fit logistic regression to X, Y.

2.3 Task 3: Validation Metrics

- Write down a list of validation metrics you learned during this course (or via any other means) that you think would be suitable for assessing how well the three methods perform at image classification. Why did you choose these?
- Write a Python function for each of these metrics. Hint: each of these functions should take a model fit and a data set and return a float or an array of floats.
- Produce a table of results that includes the three methods in the rows and the various metrics in the columns. What can you conclude from this table? Hint: make sure you use *held-out data* for this.

2.4 Bonus Task 4: Hyperparameter Selection

Only do this task if you have time left over. Most machine learning methods have *hyperparameters*, that is, variables to the algorithm or model that are not fit by the training procedure but which may effect the results nonetheless. You already saw an example: the integer K is a hyperparameter to K-means clustering.

- Using your knowledge of the three classification methods, make a short list of hyperparameters that you would like to select over. Also decide the range that these hyperparameters can reasonably take (e.g., for K-means on a small size data set of around 500 points, $K = 1, \ldots, 20$ is a suitable range, though it depends on how the data are dispersed).

[1] `http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

[2] `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`

- Choose one or several of your metrics from Section 2.3 for which you want to maximize the hyperparameters.
- Perform a *grid search* over the hyperparameters: for a suitable step size between the minimum and maximum value of each hyperparameter, try all possible combinations of hyperparameters (this is why the previous subtask ask for a *small* number of hyperparameters).
- Which hyperparameter combination works best? Can you just this in relation to the data?