Machine Learning Project Report

Valentin Herrmann, Noah Wedlich
Applied Machine Learning in Python – LMU Munich

July 15, 2025

1 Task Overview

In this project we tested if common models like kernel svms, perceptrons as well as decision tree ensembles suffer from the bias-variance tradeoff. For this purpose we trained and tested the models on simple datasets like concentric bands, interleaving half-moons, spirals and separated Gaussian clusters to produce signs of underfitting and especially of overfitting.

2 Methods

2.1 Implementation

The implementation of this project can essentially be divided into two parts: the tunable models and the samplers, both of which we will shortly describe here. We chose an object-oriented approach to allow for easy extension and reuse of the code and to integrate with the Courselib.

Tunable Model: This class provides a convenient way to create and train models on all (or a subset of) the hyperparameter combinations specified. When initialized, it receives a subclass of the Courselib `TrainableModel` class and a dictionary mapping parameter names to their possible values. When training the model, it will instantiate a model for each combination of hyperparameters and train it on the given data. Optionally, one can also provide a validator function which can limit the training to a subset of the combinations. Since the different models are independent of each other, the training is parallelized using Python's `multiprocessing` module. We also utilize queues to communicate with the main process, which allows us to display the training progress in real-time. If an optimizer is used for training, we will further wrap it to provide granular progress updates.

Samplers: The samplers are used to generate the datasets on which the models are trained. The subclasses provided here will sample from a distribution on $S \times L \subseteq [-1, 1]^2 \times \mathbb{N}$ where S is some 2D shape and L is a set of labels. The Samplers also provide methods to apply pre- and postprocessing to the data, which can be used to transform the data into a suitable format for training and to apply transformations to the data after sampling. This allows to easily test the robustness of models using domain shifts or label noise.

3 Experiments and Results

4 Discussion