

# Kernelized Linear Classification

## Project of Statistical Methods for Machine Learning

Mattia Lecchi (964860)

August 19, 2024

## 1 Dataset

The given dataset is made of 10000 examples. We first apply an affine mapping on data points by adding a new feature and setting it to 1:

$$x = (x_1, \dots, x_d) \rightarrow x' = (1, x_1, \dots, x_d) \quad (1)$$

This allow us to include the bias term in the weight vector and learn it like every other weight. After that, the dataset is split in a training set of 7000 examples and a test set of 3000 examples.

## 2 Perceptron

Figure 1 show performances of perceptron algorithm on the training and test set, based on the number of epochs. The errors on both sets are quite stable around 0.28, but it shows slightly better predictions on the very early epochs, while the error increases with the number of epochs. Looking at this results, the dataset is probably non linearly separable, as the algorithm didn't converge during the experiments and the error floats around the same values.

**Quadratic feature extraction** We try to improve the quality of predictions applying quadratic feature extraction, at the cost of a computationally more intensive predictor: starting from 11 feature, the quadratic feature extraction generate data points of 66 features. Figure 2 shows the achieved errors. In this case, despite certain spikes, the relationship between epochs and error is more evident, but it's also visible how increasing the number of epochs after a certain point brings the model to be less stable.

**Kernel trick** There's room for improvement by applying a polynomial extraction of higher degree, although the training tends to be computationally intractable even for relatively small degrees and training set, as the number of features explodes even with a relatively small degree: considering 10 features,

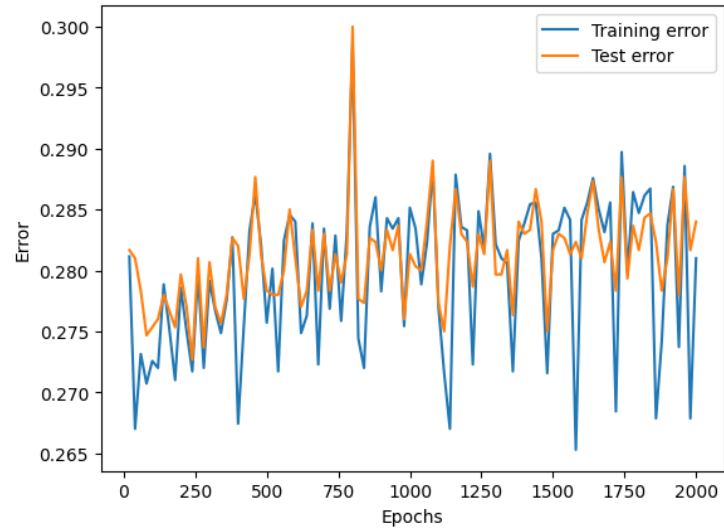


Figure 1: Training and test errors of the perceptron depending on the number of epochs

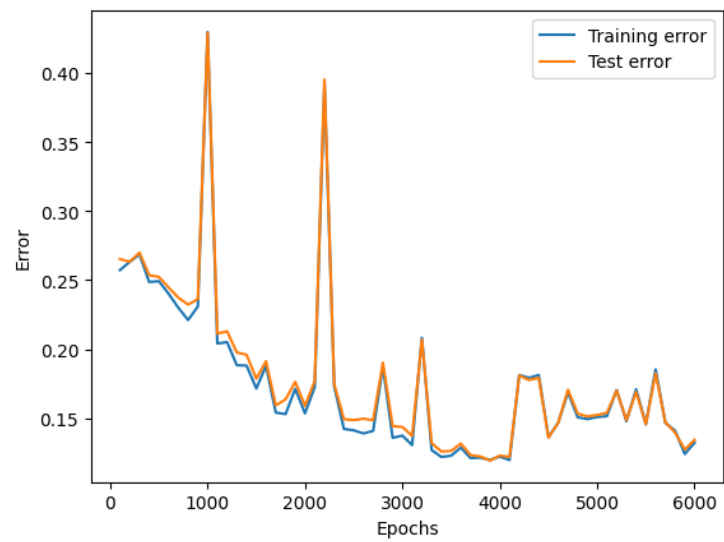


Figure 2: Training and test errors of the perceptron with quadratic feature extraction depending on the number of epochs

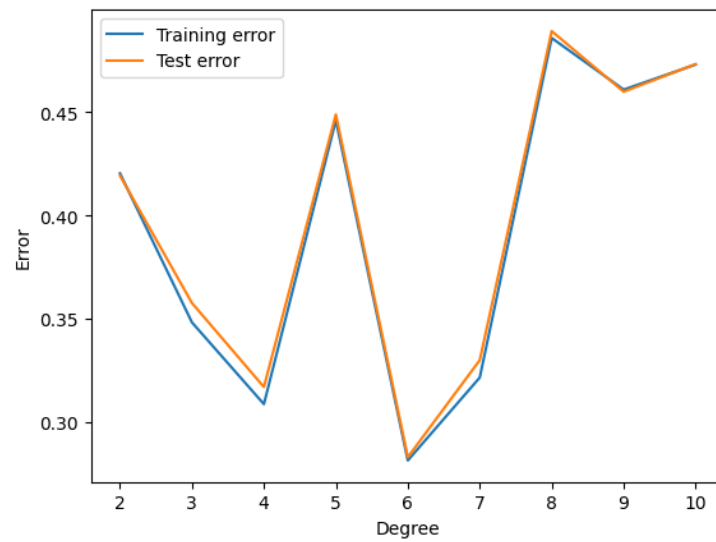


Figure 3: Training and test errors of the perceptron with quadratic feature extraction depending on the number of epochs

with a degree of 3, the number of features would be 286, while with degree set to 4 the number of features would be 1001.

Kernel methods can be used to overcome this problem. ... ..