Imperial College
London

COURSEWORK 2

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Mathematics for Machine Learning

Author:
Ion Stagkos Efstathiadis (CID: 01074696)

Date: November 16, 2020

1 Introduction

The data used throughout this coursework consist of $N = 25$, single dimension data pairs. Each data pair n is (X_n, y_n). The entirety of data can be represented concisely as vectors $\mathbf{X} \in \mathbb{R}^{N\times 1}$, $\mathbf{y} \in \mathbb{R}^{N\times 1}$. $\mathbf{X}$ is sampled uniformly from range $[0, 0.9]$. $\mathbf{y}$ is given in (1):

$$y_n = cos(10X_n^2) + 0.1sin(100X_n) \tag{1}$$

In this coursework, we try to estimate y_n for each X_n by assuming that it is normally distributed with variance σ^2 around some mean given by some function $f(X_n)$. This is represented in (2), (3):

$$y_{n,pred} = f(X_n) + \epsilon_n \tag{2}$$

$$\epsilon_n \sim N(0, \sigma^2) \tag{3}$$

Crucially, we train function $f(X_n)$ by assuming a model for it, and selecting the parameters of the model that give the maximum likelihood of data $\mathbf{y}$ stemming from the model. In each case, models are weighted sums of basis functions of order k. The basis functions are arranged appropriately in a matrix $\mathbf{\Phi} \in \mathbb{R}^{N\times(k+1)}$ for polynomial basis functions and $\mathbf{\Phi} \in \mathbb{R}^{N\times(2k+1)}$ for trigonometric basis functions. The parameters weighting them are arranged in vector $\omega \in \mathbb{R}^{(k+1)\times 1}$ for the case of polynomial basis functions and $\omega \in \mathbb{R}^{(2k+1)\times 1}$ for the case of trigonometric basis functions. Thus, it holds that:

$$\mathbf{y} = \mathbf{\Phi}\omega + \epsilon \sim N(\mathbf{0^T}, \sigma^2\mathbf{I}) \tag{4}$$

2 Question a

The maximum likelihood estimate parameters are calculated analytically using (5):

$$\omega_{MLE} = (\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}^T\mathbf{y} \tag{5}$$

This is done for 5 sets of polynomial basis functions $k \in [0, 1, 2, 3, 11]$. The fit for each order of polynomial basis functions is portrayed in Figure 1. As per Figure 1, models with $k = 0, 1, 2, 3$ underfit the data whereas the model for $k = 11$ fits the data nicely in the specified range, but rises sharply outside of the range, which renders its performance outside of the range doubtful.

3 Question b

The maximum likelihood estimate parameters are again calculated using (5). This is done for 2 sets of trigonometric basis functions $k \in [1, 11]$. The fits for the two models obtained are portrayed in Figure 2. The model with $k = 1$ underfits the data whereas that with $k = 11$ seems to overfit the data.

Figure 1: Maximum likelihood estimate fit using polynomial basis functions of order $k = 0, 1, 2, 3, 11$.

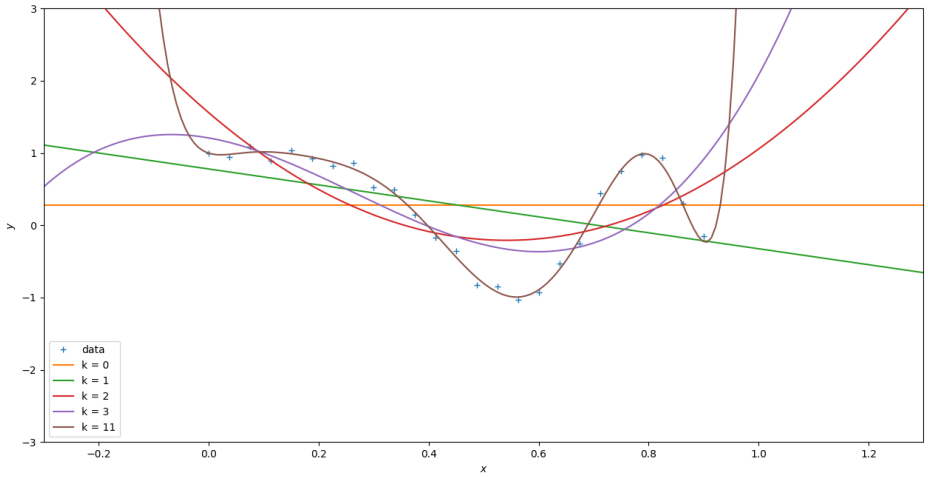


Figure 2: Maximum likelihood estimate fit using trigonometric basis functions of order $k = 1, 11$.

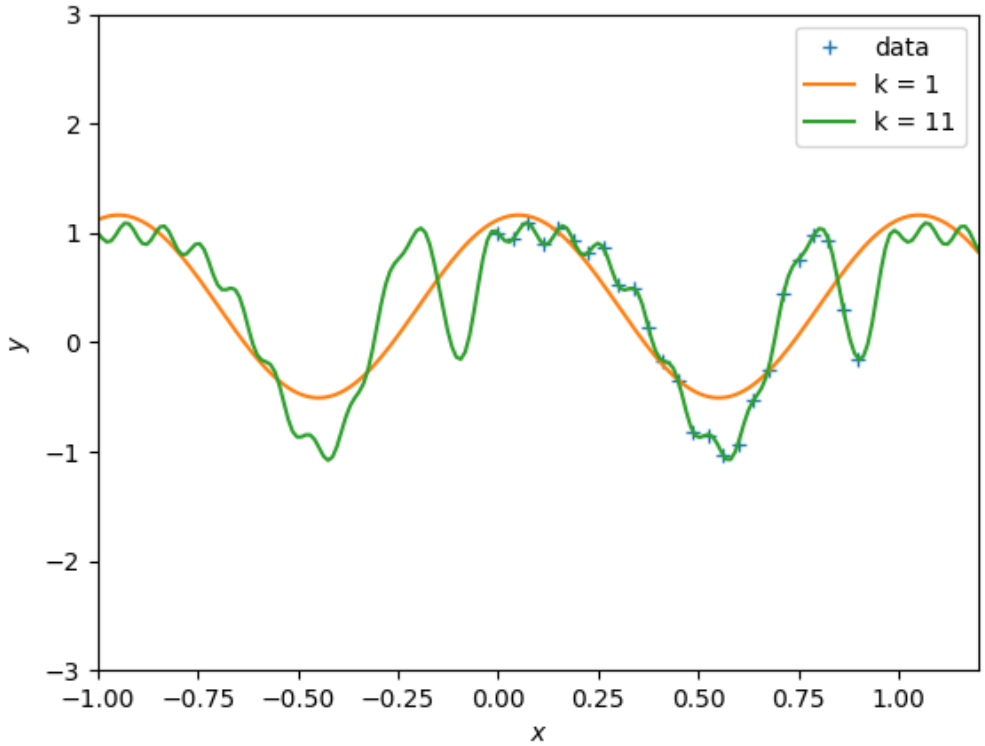
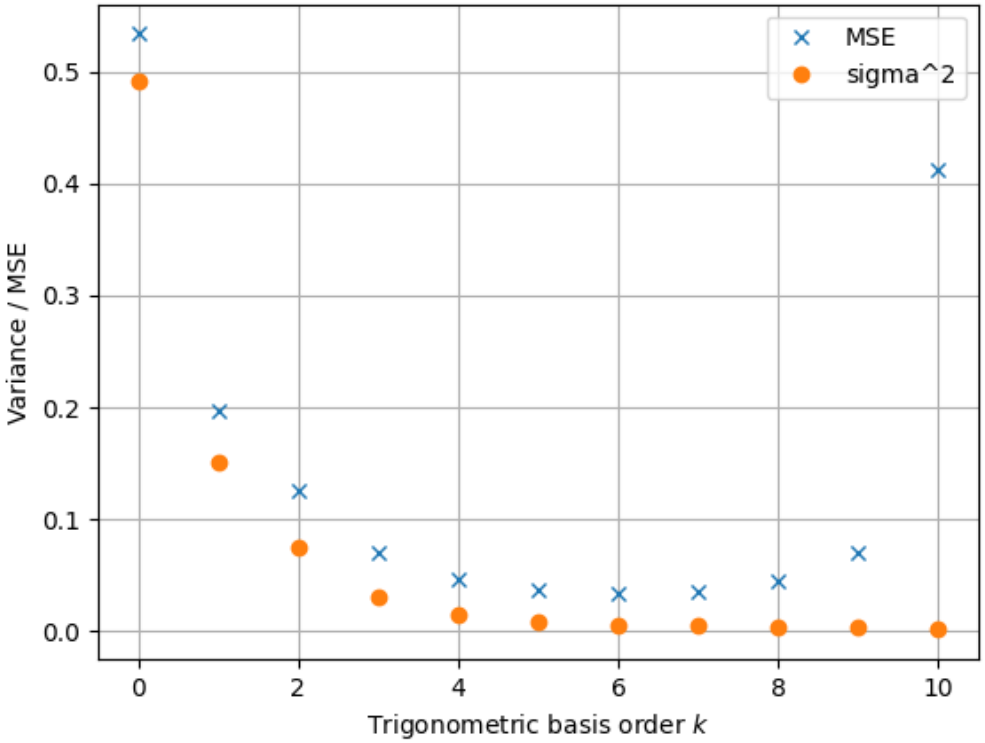


Figure 3: Variance σ^2 and MSE of leave-one-out cross-validation using trigonometric basis functions of order $k = 1to10$.



4 Questions c & d

This section investigates the effect of overfitting for models with trigonometric basis functions of order $k \in [1to10]$. Firstly, the variance of estimated values of y is plotted for models of different order, trained using all data. Secondly, the mean squared errors (MSE) between the left-out value of y and its estimated value with leave-one-out cross-validation are plotted for different models of different order.

As per Figure 3, the variance of models continually decreases with increasing order of basis functions, as expected. However, the MSE decreases up until $k = 6$ and increases thereafter. This means that models of $k > 6$ overfit the training data and perform worse against test data. The sweet spot for k that yields models which describe the data most accurately is $k = 6$.

Overfitting happens when a model is trained to reflect the idiosyncrasies of the training data that do not describe the general pattern generating the data. Therefore, when the model is tested against new data it performs worse than expected. Overfitting is prevented by introducing some metric of the complexity of the model as a hyperparameter, and using cross-validation to optimise for it. This has been done here, where this metric is the order of basis functions k. Figure 3 shows that the optimal value of k that we should use in training our models is 6.