Coursework 2
Mathematics for Machine Learning (CO-496)

Instructions

This coursework has only a writing component. You must submit the the file `write_up.pdf` for your written answers.

Data
The questions in this part will use the same 1D data. These data are pairs (x_i, y_i) where x_i are 25 values uniformly spaced in [0, 0.9], and $y_i = g(x_i)$, where

$$g(x) = \cos(10x^2) + 0.1\sin(100x).$$

This data set can be generated as follows:

```
import numpy as np
N = 25
X = np.reshape(np.linspace(0, 0.9, N), (N, 1))
Y = np.cos(10*X**2) + 0.1 * np.sin(100*X)
```

Basis Functions

We will use the following classes of basis functions $\boldsymbol{\phi}(x) = (1, \phi_1(x), ..., \phi_J(x))^T$ where $J + 1$ is the dimension of the basis functions.

- Polynomial of degree K:
$$\phi_j(x) = x^j,$$
for $j = 1, 2, \ldots, K$.

- Trigonometric of degree K with unit frequency:
$$\phi_{2j-1}(x) = \sin(2\pi jx)$$
$$\phi_{2j}(x) = \cos(2\pi jx)$$
for $j = 1, 2, \ldots, K$

- Gaussian with scale l and means μ_j:
$$\phi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2l^2}\right)$$

The 1 in the first position is to absorb the bias term into the weights. This simplifies the notation, but note that Kth degree polynomial basis functions are of dimension $K + 1$, Kth degree trigonometric basis functions are of dimension $2K + 1$, and the Gaussian basis functions with K means are of dimension $K + 1$. Another useful notation is the $N \times M$ design matrix, defined as $(\mathbf{\Phi})_{nm} = (\boldsymbol{\phi}_m(x_n))$, where $n = 1, 2..., N$ indexes the data points and $m = 1, 2, ..., M$ indexes the basis functions.

1 Linear Regression

In this question we consider a factorizing likelihood

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p(y_i|x_i)$$

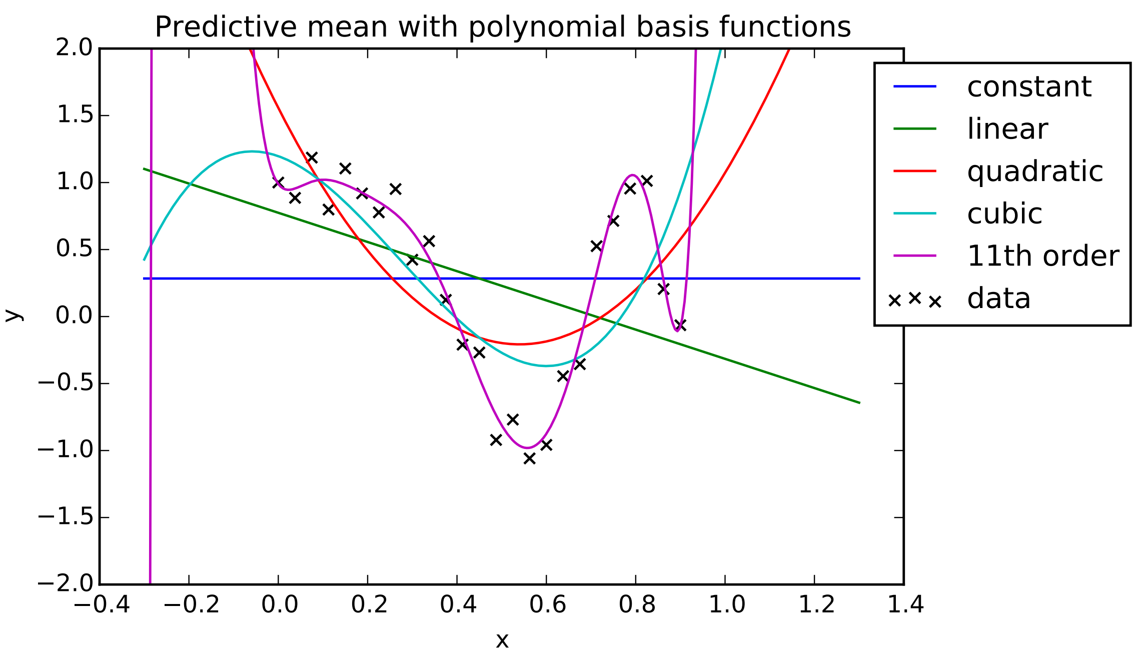
and Gaussian linear model

$$y_i \sim \mathcal{N}(\boldsymbol{w}^T \boldsymbol{\phi}(x_i), \sigma^2)$$

with basis functions as defined in the instructions section. We will often be changing the basis functions so all your derivations should be in terms of $\boldsymbol{\phi}$.

In this question we will find the maximum likelihood solution for the parameters, conditioned on the data. The data is defined in instruction section.

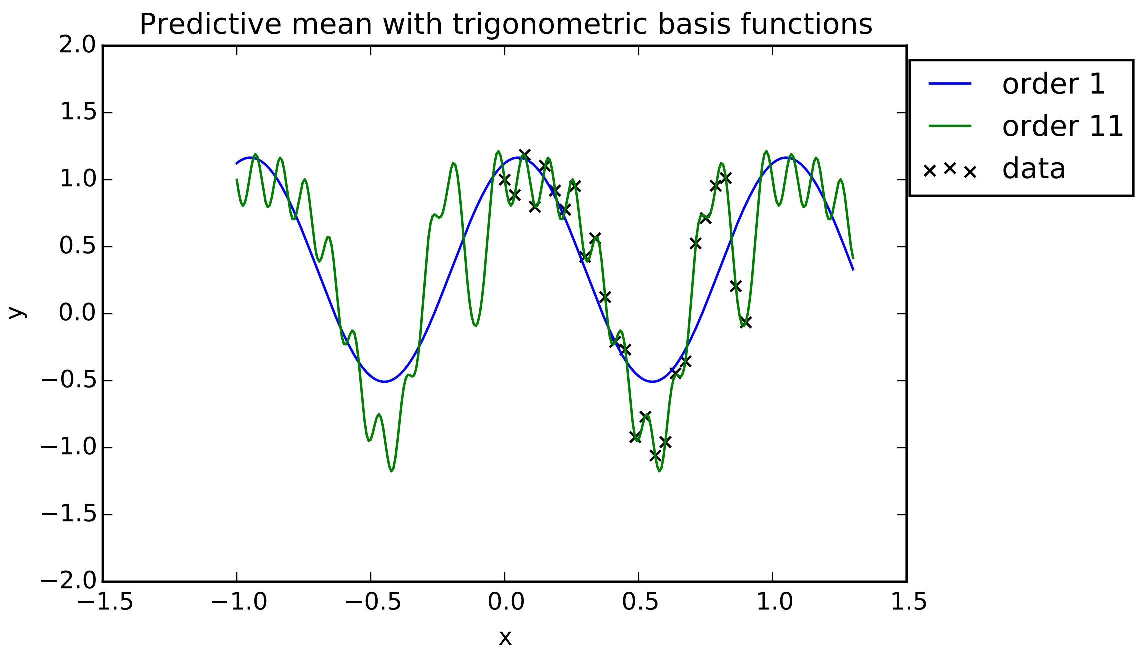
a) By first finding the maximum likelihood solution for the parameters σ^2 and $\boldsymbol{w}$ in terms of $\boldsymbol{\Phi}$, plot the predictive mean at test points in the interval $[-0.3, 1.3]$ in the case of polynomial basis functions of order 0, 1, 2, 3 and also order 11. Plot all the curves on the same axes, showing also the data.



- One plot with two lines and three curves, clearly labeled (use colour and don't bother making it readable in black and white) with legend sensibly positioned, including the data. You will need to set the y-axis limits to make the plot clear.
- Smooth curves (200 uniformly spaced points should do)

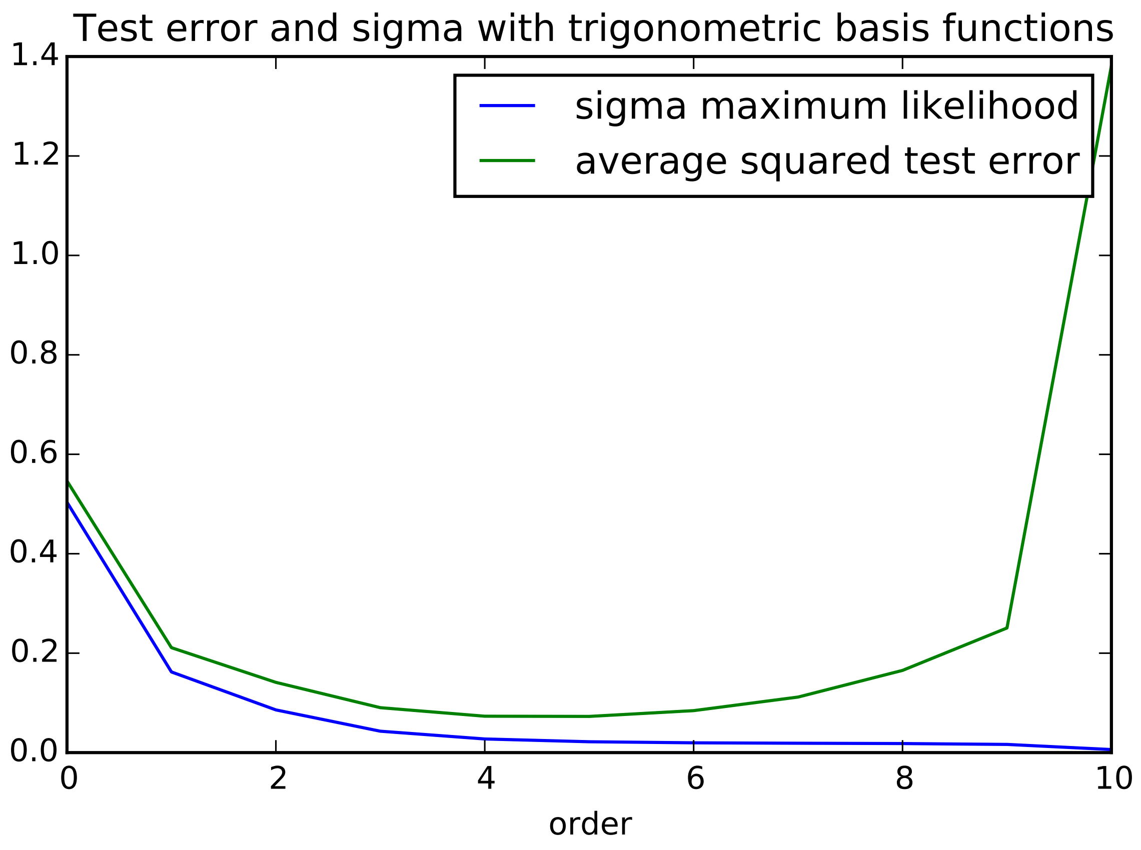
- Correct order 0 and 1
- Correct order 2 and 3
- Correct order 11

b) Repeat the previous part but this time with trigonometric basis functions of orders 1 and 11. Use test points in [-1, 1.2] to see the periodicity. Note that your basis functions should be of size $2J + 1$ for order J (don't forget the bias term)



- One graph with two curves, clearly labeled
- Correct order 1
- Order 11 approximately correct
- Order 11 correct

c) In this part you will investigate over-fitting with leave-one-out cross validation. You should use trigonometric basis functions of order 0 to 10 inclusive and for each choice use leave-one-out cross validation to estimate the average squared test error. Plot this average error on a graph against order of basis together. On the same graph plot also the maximum likelihood value for σ^2.



- Curve for σ^2_{ML}, approximately correct
- Curve for test squared error, approximately correct
- Both correct
- Labeled and sensibly formatted

d) Briefly describe the concept of over-fitting, using your graph in the previous part as an illustrative example. You should also refer to your plots from the first two parts of this question.

σ_{ML} always decreases with more parameters. This is shown in the graph. This is because the model with more parameters has more flexibility to fit to the data and so can reduce the average squared error on the training data.

However, the test error increases after a certain point. This can be seen in the graph from about order 5. This is because as the model becomes more flexible it fits more closely to the test data but does not generalize to unseen data. The plots in parts a) and b) also illustrate this: with high order basis functions the predictive mean passes very close to the test points, but it becomes increasingly erratic.

- A point about the shape of the σ^2_{ML} graph with some explanation

- A point about the shape of the test error graph with some explanation
- A high level description of over-fitting
- Comments made with reference to the graphs from previous parts (i.e. not generic explanations)

2 Ridge Regression

a) A non-probabilistic approach to linear regression is to set $y_i = \boldsymbol{w}^T\boldsymbol{\phi}(x_i)$ and then find the best $\boldsymbol{w}$ by minimizing some loss function. Show that linear regression with regularized least squares loss function

$$L(\boldsymbol{w}) = \sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(x_i))^2 + \lambda\sum_{j=1}^{M} w_j^2$$

is equivalent to the MAP estimate for $\boldsymbol{w}$ with the factorized Gaussian likelihood $y_i \sim \mathcal{N}(\boldsymbol{w}^T\boldsymbol{\phi}(x_i), \sigma^2)$ and a certain prior for $\boldsymbol{w}$. You must state the prior you place on $\boldsymbol{w}$ and explicitly connect the parameters of your prior and the parameters of the likelihood to λ. Explain also the intuition behind this loss function.

The prior is

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha\boldsymbol{I})$$

With this prior the log likelihood is

$$-\frac{N}{2}\log 2\pi\sigma^2 - \frac{M}{2}\log 2\pi\alpha - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}\boldsymbol{\phi}(x_i))^2 - \frac{1}{2\alpha}\sum_{j=1}^{M} w_j^2$$

The additive constants independent of $\boldsymbol{w}$ are irrelevant, and we can multiply by a constant factor without changing the solution for $\boldsymbol{w}$. If we multiply by $2\sigma^2$ and drop the constants we see that the two expressions are identical(apart from negative sign) with

$$\lambda = (1/\alpha)\sigma^2$$

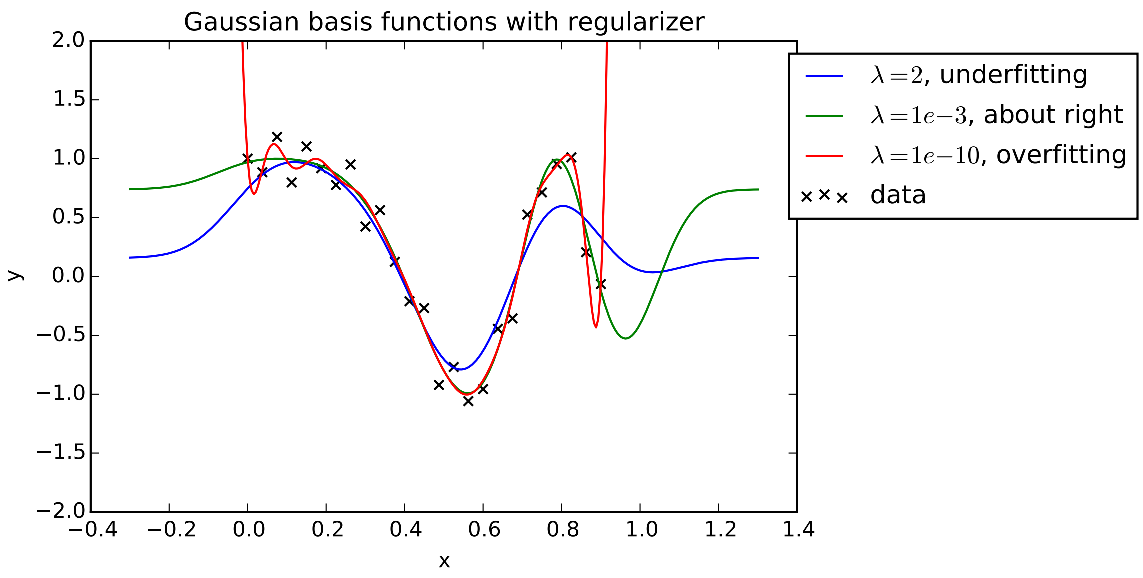
So the maximum posterior point is equivalent to the minimum loss of the regularized least squares model.

The quadratic penalty term avoids the parameters getting too large. This makes sense in this linear model as very large parameter values correspond to great sensitivity between input and output values, which is likely to be undesirable and lead to over-fitting.

- Statement of prior

- Statement of (negative) log likelihood including the log terms
- either differentiation of both to show the solution for $\boldsymbol{w}$ are the same (hint: you may find it easier to work with the expression if you use vector form i.e. $\boldsymbol{w}^T\boldsymbol{w}$ for the second term), or else an argument to show equivalence of solution.
- Statement of relationship between constants
- Explanation of intuition of the loss function, with sensible discussion of the effects of the regularizer and why this might be beneficial

b) Find the regularized least squares value for $\boldsymbol{w}$ using 20 Gaussian basis functions of scale 0.1 with means equally spaced in $[0, 1]$ and plot the regression function for test points between -0.3 and 1.3 for three values of λ of your choosing. Your chosen values of λ should illustrate under-fitting, over-fitting and somewhere more satisfactory in between. Make sure you label which is which, together with the values of λ that you used.



- One curve showing under-fitting, with value of λ stated
- One curve showing over-fitting, with value of λ stated
- One curve showing a reasonable fit, with value of λ stated

c) Optional: We now consider linear regression from a frequentist point of view (i.e. we consider the randomness to be on the dataset rather than the parameters). To this end we will create new datasets by replacing the second high frequency term by a Gaussian noise term of the same variance. The true function we are trying to model is then $y = \cos(10x^2)$. To generate such datasets in python you can use something like

```
def make_sample_dataset():
    N = 25
    X = np.reshape(np.linspace(0, 0.9, N), (N, 1))
    noise = np.random.randn(*X.shape)
    Y = np.cos(10*X**2)+ 0.2 * (2**-0.5) * noise
    return X, Y
```

We consider the inputs x_i as fixed, so now we know the 'correct' answer is $y_i^{true} = \cos(10x_i^2)$. We denote the predicted output values from our model, given a particular dataset $\mathcal{D}^d$, as y_i^d, where the D datasets are indexed by d. If we take an average over a large number of datasets we might hope that the empirical average $y_i^{avg} = \frac{1}{D}\sum_{d=1}^{D} y_i^d$ tends to the true value y_i^{true}. Our model is not guaranteed to achieve this, however, as the model may be too simple to capture the true underlying function.

To see how close our estimator comes to the true value we can consider the average squared error from the true values, E, where the average is taken over the datasets:

$$E = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{N}\sum_{i=1}^{N}(y_i^d - y_i^{true})^2$$

i) Show that we can write E as

$$E = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{N}\sum_{i=1}^{N}(y_i^d - y_i^{avg})^2 + \frac{1}{N}\sum_{i=1}^{N}(y_i^{avg} - y_i^{true})^2 \tag{1}$$

Identify the terms as 'bias (squared)' and 'variance' and explain how (1) can be interpreted as a bias variance trade-off.

We can write

$$\begin{aligned} E =& \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}(y_i^d - y_i^{avg} + y_i^{avg} - y_i^{true})^2 \\ =& \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}\left[(y_i^d - y_i^{avg})^2 - 2(y_i^d - y_i^{avg})(y_i^{avg} - y_i^{true}) + (y_i^{avg} - y_i^{true})^2\right] \end{aligned}$$

The middle term vanishes due to the definition of y_i^{avg} and the last term is independent of d so drops out of the sum losing a factor of D along the way.

The first term is the variance as it is is the average squared error of the predictions about their average value. A small value of this term indicates that all data sets give the same predictions. A large value means that the predictions are very different on different data sets i.e. you cannot trust a particular answer on one data set.

The second term is the bias (squared) as it is the average squared error of the average value from the true value. A small value of this term indicates that the average predictions are close to the true values. A large value indicates that the predictions differ from the true predictions i.e. you never learn the function even given lots of data.

The overall error is a sum of these two positive terms. If there is noise in the data we cannot have both close to zero. If the model has very low bias i.e. it can capture the underlying function then it must have high variance. If the model has very low variance then it must be biased so is insufficiently flexible to capture the underlying function.

This question is optional.

- Some manipulation to required answer

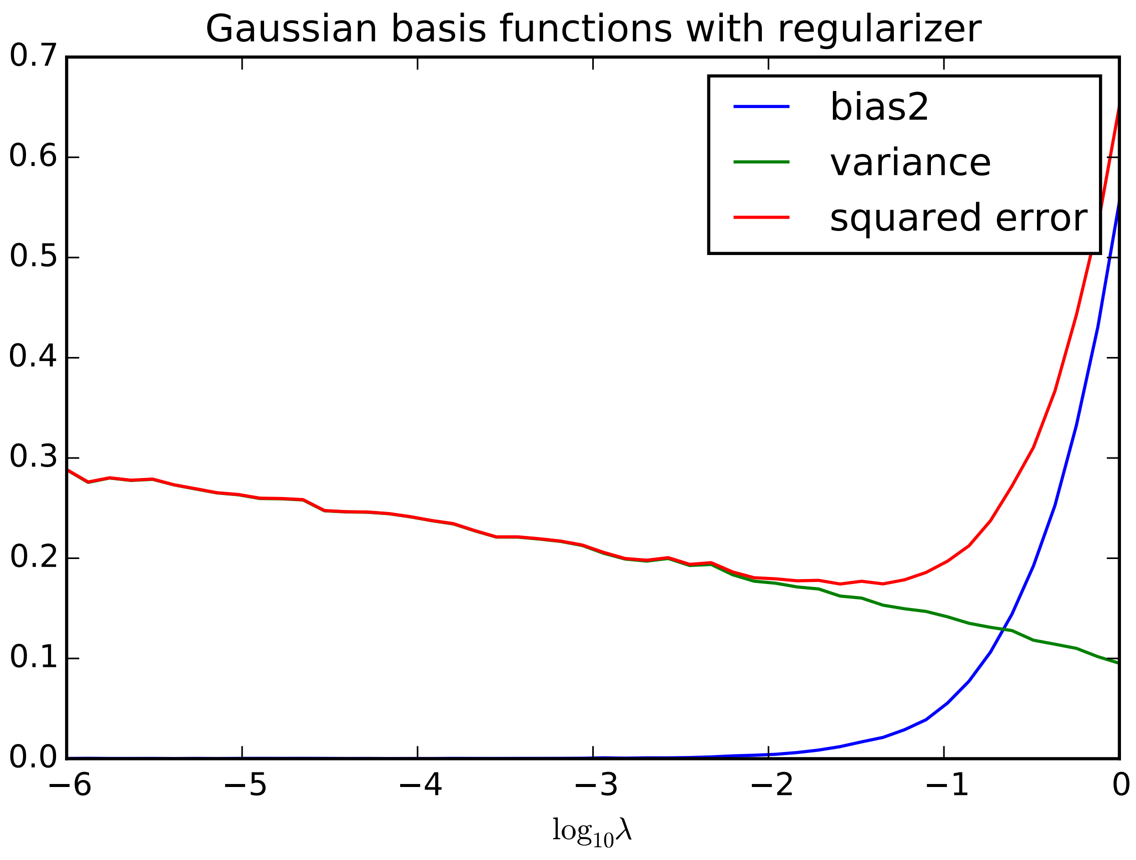
- Identification of the two terms

- Discussion of bias and what small and large values

- Discussion of variance and what mall and large values s

- Discussion of the trade off when modeling noisy data.

ii) Demonstrate the effect of regularization on the bias-variance trade-off by generating multiple datasets as described above with the regularized linear model with 20 Gaussian basis functions of scale 0.1 equally spaced in [0, 1]. Plot a graph of the bias squared, variance and squared squared error against the regularization parameter.



This question is optional.

- Reasonably smooth curves (1000 datasets should do)
- Bias curve correct
- Variance curve correct
- Scales suitable chosen (log scale for the regularizer). Minimum of total squared error clear. Legend clear.