Mathematics for Machine Learning
Coursework 3 - Bayesian Linear Regression

Stanislas Hannebelle

November 2018

1 Question A: Log-Marginal-Likelihood and Gradient

Let $ML_{\Phi,y}$ be the function associating (α, β) to the marginal likelihood.
Here is the analytic expression of $ML_{\Phi,y}$ which enables us to implement the lml python function:

$$ML_{\Phi,y} = \frac{-N\log(2\pi) - \log(\det(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)) - y^T(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}y}{2}$$

Also, thanks to derivative calculus, we can compute both components of the gradient of $ML_{\Phi,y}$:

$$\frac{\partial ML_{\Phi,y}(\alpha,\beta)}{\partial\alpha} = \frac{y^T(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}\Phi\Phi^T(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}y - tr((\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}\Phi\Phi^T)}{2}$$

$$\frac{\partial ML_{\Phi,y}(\alpha,\beta)}{\partial\beta} = \frac{y^T(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}(\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1}y - tr((\alpha\Phi\Phi^T + \beta\mathbb{I}_N)^{-1})}{2}$$

Please check the python implementation of this functions in answers.py.

2 Question B: Maximisation of LML for Linear Basis Functions

By observing contour plots of functions $ML_{\Phi,y}$ on several intervals, we observe that this function presents a maximum in the interval $(\alpha,\beta) \in [0.1, 3] \times [0.3, 1.1]$. So, (1.0,1.0) could be a good starting point for our gradient descent as it is not to far from the maximum.
We will use a constant step-size for this gradient descent. We observe that, if we use a step-size greater than 0.1, the gradient descent algorithm diverges. Also, if we use a gradient descent lower than 0.01, the gradient descent algorithm needs to many iterations to converge. Finally, if we set the step-size at 0.025,

the gradient descent algorithm converges to a maximum after 368 iterations. The maximum equals approximately -27.6087946 and the corresponding alpha and beta values are:

$$\alpha^* = 0.42455543$$

and,

$$\beta^* = 0.44923131$$

Here is a contour plot of function $ML_{\Phi,y}$ illustrating the converging gradient descent algorithm.

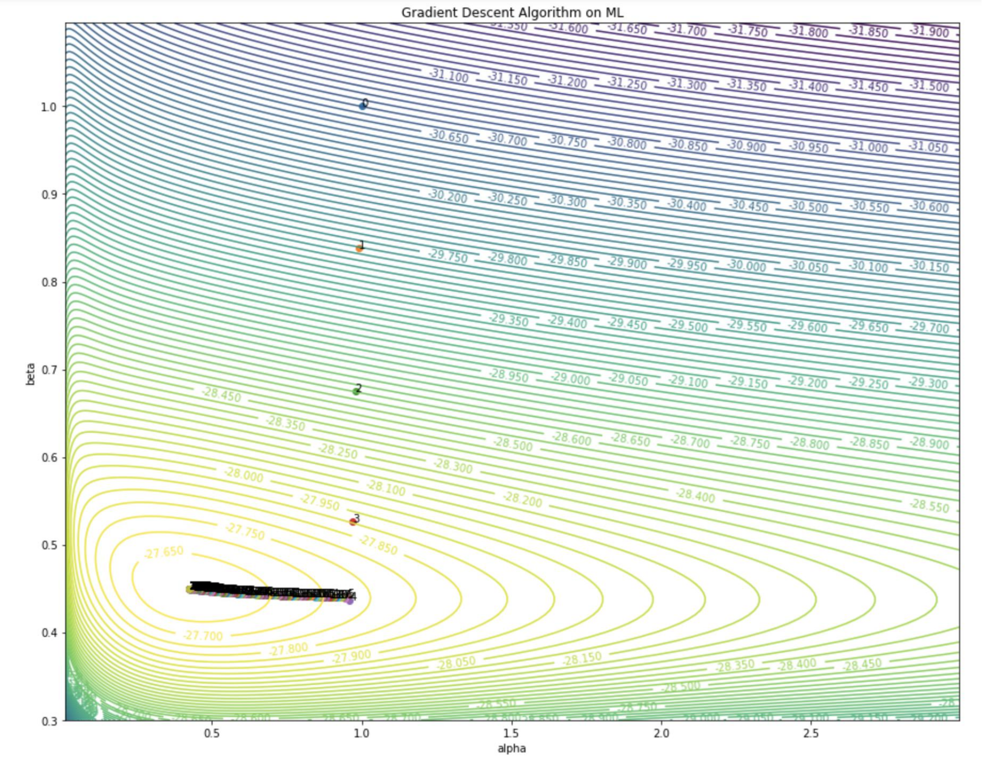
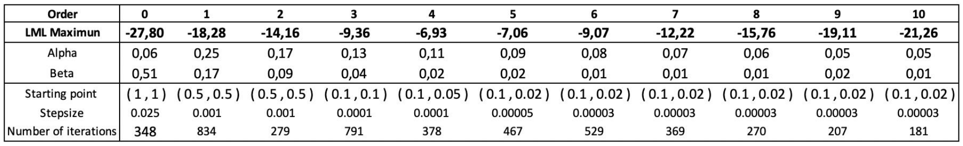


Figure 1: Contour Plot of $ML_{\Phi,y}$ and Iterations of Gradient Descent of step-size 0.025 starting at (1.0,1.0)

3 Question C: Case of Trigonometric Basis Functions

As the instructor Mr. Salimbeni said on Piazza, order 11 is near singular so we will only consider orders from 0 to 10 inclusive.
Here is a table showing corresponding results of gradient descents.



Order	0	1	2	3	4	5	6	7	8	9	10
LML Maximun	**-27,80**	**-18,28**	**-14,16**	**-9,36**	**-6,93**	**-7,06**	**-9,07**	**-12,22**	**-15,76**	**-19,11**	**-21,26**
Alpha	0,06	0,25	0,17	0,13	0,11	0,09	0,08	0,07	0,06	0,05	0,05
Beta	0,51	0,17	0,09	0,04	0,02	0,02	0,01	0,01	0,01	0,02	0,01
Starting point	(1 , 1)	(0.5 , 0.5)	(0.5 , 0.5)	(0.1 , 0.1)	(0.1 , 0.05)	(0.1 , 0.02)	(0.1 , 0.02)	(0.1 , 0.02)	(0.1 , 0.02)	(0.1 , 0.02)	(0.1 , 0.02)
Stepsize	0.025	0.001	0.001	0.0001	0.0001	0.00005	0.00003	0.00003	0.00003	0.00003	0.00003
Number of iterations	348	834	279	791	378	467	529	369	270	207	181

Figure 2: Results of Gradient Descent Algorithm over orders from 0 to 10

Let's plot the log-marginal-likelihood maximum versus the order of the trigonometric functions.

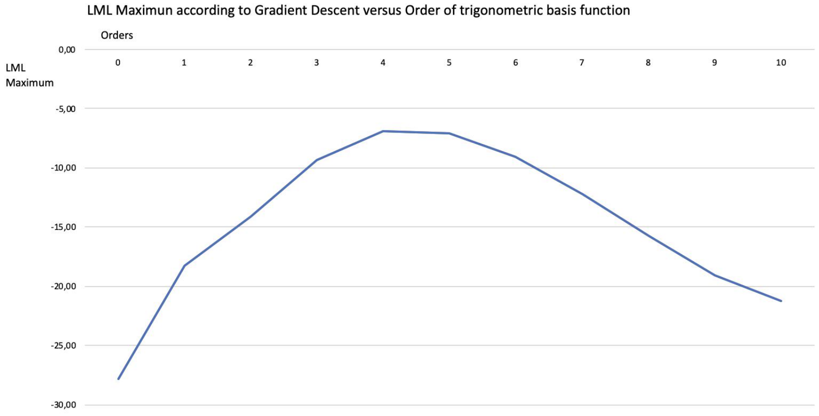


Figure 3: Log-Marginal-Likelihood Maximum versus the Order of the Trigonometric Basis Functions

To be able to compare the Bayesian approach to the cross validation approach, let's present the plot corresponding to the cross validation approach that comes from the second coursework.

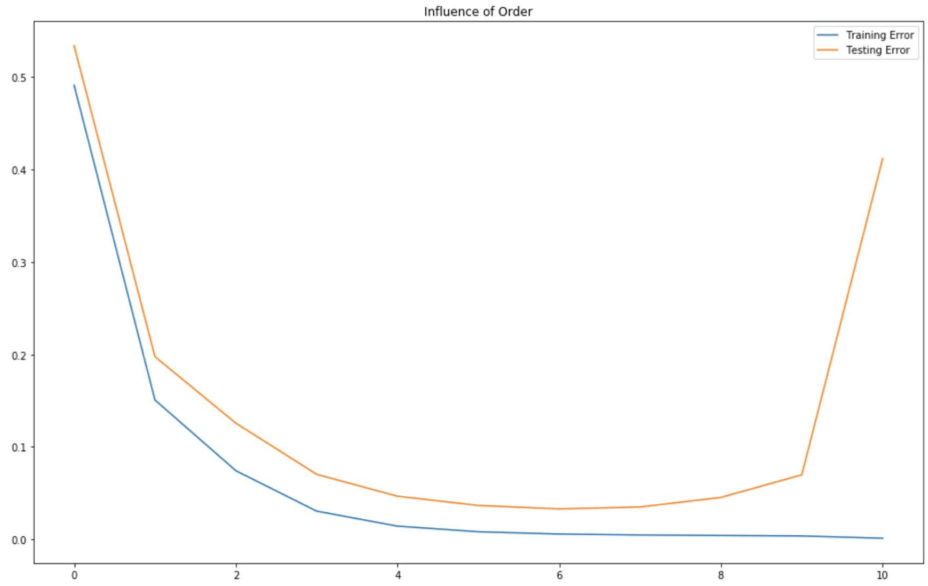


Figure 4: Impact of the order of a trigonometric basis function linear regression over training and testing errors during a leave-one-out cross validation

Firstly, we can observe that with the Bayesian method the optimal order is 4. 5 could also be a good order choice in this case. However, in the cross validation approach good order choice would be 5 or 6.

The main point in favor of the cross validation approach compared to the Bayesian approach is the simplicity of the mathematical calculus. Indeed, the log-likelihood expression is very simple compared to the Bayesian case in which we need to compute the integral of a product of Gaussian functions.

However, Bayesian linear regression ensures more accurate results as we consider the whole range of inferential solutions.

4 Question D

Here, the posterior distribution over the weights is:
$\mathcal{N}((\frac{1}{\alpha}\mathbb{I}_M + \frac{1}{\beta}\Phi^T\Phi)^{-1}(\frac{1}{\beta}\Phi^T y), (\frac{1}{\alpha}\mathbb{I}_M + \frac{1}{\beta}\Phi^T\Phi)^{-1})$
Thanks to this expressions, we can sample five sets of weights and draw the requested graph:

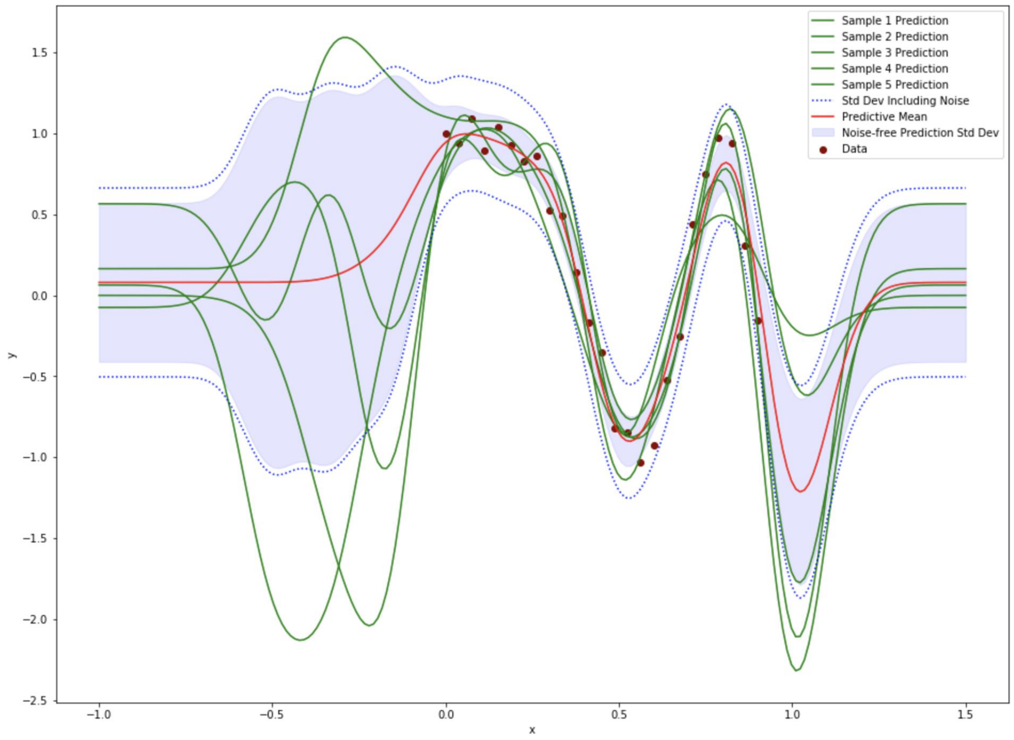


Figure 5: Predictive Means and Errors with predictions associated to 5 different samples.