Mathematics for Machine Learning
Coursework 2

Stanislas Hannebelle

October 2018

1 Linear Regression

1.1 Question a

Let's minimize the opposite of the log-likelihood function. We will note $F(\sigma^2, \mathbf{w})$ this function that we have to minimize.
Let σ_0^2 and $\mathbf{w}_0$ be the solution parameters of the maximum likelihood.
Then,

$$\sigma_0^2 = \arg\min_{\sigma^2} F(\sigma^2, \mathbf{w})$$

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}} F(\sigma^2, \mathbf{w})$$

1.1.1 σ_0^2 computation

By definition of F,

$$F(\sigma^2, \mathbf{w}) = -log(p(y|x))$$

$$F(\sigma^2, \mathbf{w}) = -log(\prod_{i=1}^{N} p(y_i|x_i))$$

Then,

$$F(\sigma^2, \mathbf{w}) = -\sum_{i=1}^{N} log(p(y_i|x_i))$$

Yet,

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

So,

$$F(\sigma^2, \mathbf{w}) = -\sum_{i=1}^{N} log(\frac{\exp -\frac{(y_i - w^T \phi(x_i))^2}{2\sigma^2}}{\sigma\sqrt{2\pi}})$$

$$F(\sigma^2, \mathbf{w}) = -\sum_{i=1}^{N} log(\frac{\exp -\frac{(y_i - w^T \phi(x_i))^2}{2\sigma^2}}{\sigma\sqrt{2\pi}})$$

Then,

$$F(\sigma^2, \mathbf{w}) = Nlog(\sigma\sqrt{2\pi}) + \sum_{i=1}^{N} \frac{(y_i - w^T \phi(x_i))^2}{2\sigma^2}$$

So,

$$F(\sigma^2, \mathbf{w}) = Nlog(\sqrt{2\pi\sigma^2}) + \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma^2}$$

$$F(\sigma^2, \mathbf{w}) = \frac{N}{2} log(2\pi) + \frac{N}{2} log(\sigma^2) + \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma^2}$$

Let's derive this function with respect to σ^2:

$$\frac{\partial F(\sigma^2, \mathbf{w})}{\partial \sigma^2} = \frac{N}{2\sigma^2} - \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma^4}$$

Yet, this partial derivative equals 0 when σ^2 equals σ_0^2.
So,

$$\frac{N}{2\sigma_0^2} - \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma_0^4} = 0$$

Thus,

$$\sigma_0^2 = \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{N}$$

So, σ_0^2 equals the training quadratic error.

1.1.2 $\mathbf{w}_0$ computation

As we saw,

$$F(\sigma^2, \mathbf{w}) = \frac{N}{2} log(2\pi) + \frac{N}{2} log(\sigma^2) + \frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma^2}$$

Let's derive this function with respect to $\mathbf{w}$:

$$\frac{\partial F(\sigma^2, \mathbf{w})}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left(\frac{(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T(\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})}{2\sigma^2} \right)$$

$$\frac{\partial F(\sigma^2, \mathbf{w})}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left(\frac{\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\Phi(\mathbf{x})\mathbf{w} + \mathbf{w}^T\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}}{2\sigma^2} \right)$$

$$\frac{\partial F(\sigma^2, \mathbf{w})}{\partial \mathbf{w}} = \frac{-2\Phi(\mathbf{x})^T\mathbf{y} + 2\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}}{2\sigma^2}$$

Yet, this partial derivative equals 0 when $\mathbf{w}$ equals $\mathbf{w}_0$. So,

$$-2\Phi(\mathbf{x})^T\mathbf{y} + 2\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}_0 = 0$$

Thus,

$$\Phi(\mathbf{x})^T\mathbf{y} = \Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}_0$$

So, as $\Phi(\mathbf{x})^T\Phi(\mathbf{x})$ is invertible,

$$\mathbf{w}_0 = (\Phi(\mathbf{x})^T\Phi(\mathbf{x}))^{-1}\Phi(\mathbf{x})^T\mathbf{y}$$

1.1.3 Illustration of the predictive mean

Let μ_0 be the maximum likelihood predictive mean of $x \in \mathbb{R}$.
Then,

$$\mu_0(x) = \mathbf{w}_0^T \phi(x)$$

So,

$$\mu_0(x) = ((\Phi(\mathbf{x})^T \Phi(\mathbf{x}))^{-1} \Phi(\mathbf{x})^T \mathbf{y})^T \phi(x)$$

Let's plot the value of this mean in the case of polynomial basis functions of order 0, 1, 2, 3 and 11.

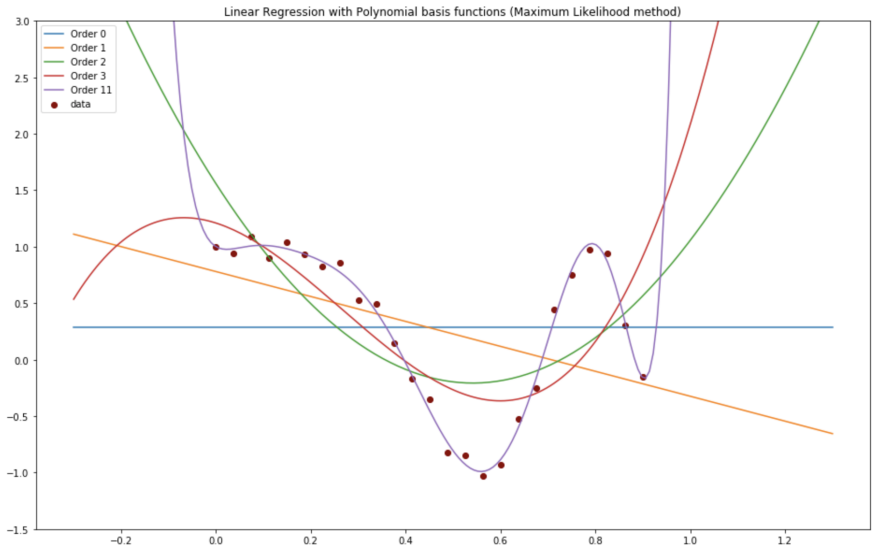

Figure 1: Data and Linear Regressions with Polynomial basis functions of order 0, 1, 2, 3 and 11 thanks to the Maximum Likelihood method

Remark: On this graph, we can observe that orders 0, 1, 2 and 3 under-fit the data as the training errors look high. Also, order 11 seems to over-fit the data as training errors is low but the predictions diverges quickly towards $+\infty$ outside of [0,1]. A choice of order between 4 and 10 would probably be better to fit the data.

1.2 Question b

Thanks to the same method as in the previous section, we can get the following graph.

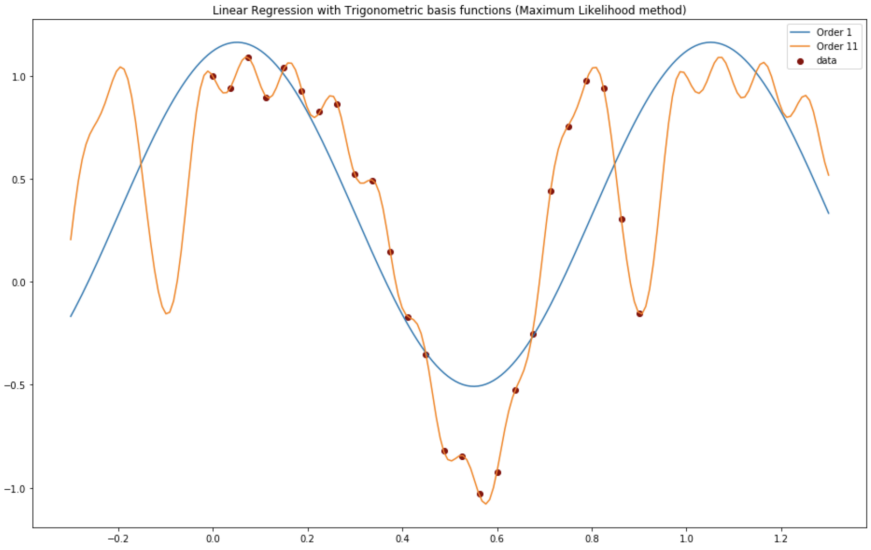

Figure 2: Data and Linear Regressions with Trigonometric basis functions of order 1 and 11 thanks to the Maximum Likelihood method

Remark: Here too, we can observe that order 1 under-fits the data whereas order 11 probably over-fits the data.

1.3 Question c

As we previously saw, σ_0^2 (or σ_{ML}^2 in the subject pdf) corresponds to the training error.
Thanks to a leave-one out cross validation, let's plot the impact of the order over the training and testing errors to illustrate over-fitting.

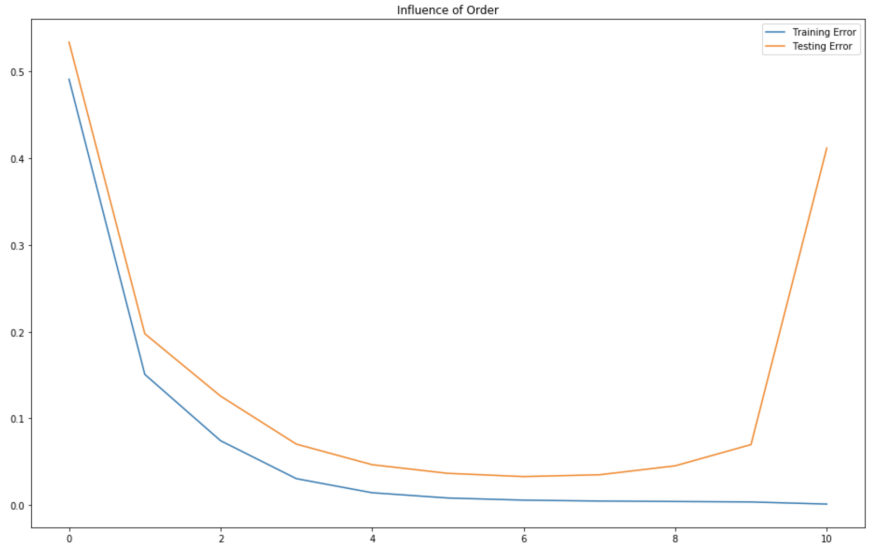

Figure 3: Impact of the order of a trigonometric basis function linear regression over training and testing errors during a leave-one-out cross validation

1.4 Question d

Firstly, let's observe the shape of the training error ($=\sigma_{ML}^2$) on Figure 3. We can see that it is decreasing when the order increases. This is intuitive. Indeed, as the order increases, our model can use more complex functions, therefore, it is easier to predict the training data and the training error decreases.
Secondly, let's observe the shape of the testing error on Figure 3. We can see that it is firstly decreasing from order 0 to 6. At this stage, complexifying the model by increasing the order, enhances the model. However, from order 6 to 10, the model over-fits the data as we see the testing error increasing.
Over-fitting qualifies a model that has extracted information about the noise contained in the training data. This information leads to increasing the testing error. Over-fitting happens when the model is free to use complex functions. To avoid over-fitting, simple model should be used as these model cannot see residual variations that are the cause of over-fitting.
In our previous studies, choosing a simple model correspond to choosing a low

order.
The orange curve of Figure 2 illustrates over-fitting as we can clearly see that the model consider residual variations of the training data (we can see a lot of small waves.). An order 11 is here to high, it is the cause of over-fitting.
Figure 3 from order 6 to 10 is characteristic of over-fitting. Indeed, the testing error increase whereas the training errors decreases when the complexity of the model increases.
Finally, as we saw in question a, the divergence of the order 11 curve outside of [0,1] is probably due to over-fitting.

2 Ridge Regression

2.1 Question a

2.1.1 Minimizing $L(\mathbf{w})$

By definition,

$$\mathbf{w}_{ridge} = \arg\min_{\mathbf{w}} L(\mathbf{w})$$

and,

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

So,

$$L(\mathbf{w}) = (\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T (\mathbf{y} - \Phi(\mathbf{x})\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}$$

So,

$$L(\mathbf{w}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\Phi(\mathbf{x})\mathbf{w} + \mathbf{w}^T\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w} + \lambda\mathbf{w}^T\mathbf{w}$$

Let's derive L with respect to $\mathbf{w}$,

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -2\Phi(\mathbf{x})^T\mathbf{y} + 2\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w} + 2\lambda\mathbf{w}$$

Yet, this partial derivative equals 0 when $\mathbf{w}$ equals $\mathbf{w}_{ridge}$.
Thus,

$$-2\Phi(\mathbf{x})^T\mathbf{y} + 2\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}_{ridge} + 2\lambda\mathbf{w}_{ridge} = 0$$

So,

$$(\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \lambda\mathbb{I}_M)\mathbf{w}_{ridge} = \Phi(\mathbf{x})^T\mathbf{y}$$

As $(\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \lambda\mathbb{I}_M)$ is invertible ($\lambda > 0$), we can conclude that:

$$\mathbf{w}_{ridge} = (\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \lambda\mathbb{I}_M)^{-1}\Phi(\mathbf{x})^T\mathbf{y}$$

2.1.2 Solving MAP

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} -log(p(\mathbf{w}|\mathbf{y}, \mathbf{x}))$$

So, thanks to the Bayes theorem,

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} -log(p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w}))$$

Then,

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} -log(p(\mathbf{y}|\mathbf{x}, \mathbf{w})) - log(p(\mathbf{w}))$$

As we previously saw,

$$-log(p(\mathbf{y}|\mathbf{x}, \mathbf{w})) = \frac{\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\Phi(\mathbf{x})\mathbf{w} + \mathbf{w}^T\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}}{2\sigma^2}$$

Then,

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} \frac{-2\mathbf{y}^T\Phi(\mathbf{x})\mathbf{w} + \mathbf{w}^T\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}}{2\sigma^2} - log(p(\mathbf{w}))$$

Let's place a Gaussian prior on $\mathbf{w}$: $\mathbf{w} \sim \mathcal{N}(0, b^2)$
Then,

$$-log(p(\mathbf{w})) = -log(\prod_{i=1}^{M} \frac{\exp -\frac{w_i^2}{2b^2}}{\sqrt{2\pi b^2}}$$

Then,

$$-log(p(\mathbf{w})) = -\sum_{i=1}^{M} log(\frac{\exp -\frac{w_i^2}{2b^2}}{\sqrt{2\pi b^2}}$$

So,

$$-log(p(\mathbf{w})) = \frac{Mlog(2\pi b^2)}{2} + \sum_{i=1}^{M} \frac{w_i^2}{2b^2}$$

So,

$$-log(p(\mathbf{w})) = \frac{Mlog(2\pi b^2)}{2} + \frac{\mathbf{w}^T\mathbf{w}}{2b^2}$$

Thus,

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} \frac{-2\mathbf{y}^T\Phi(\mathbf{x})\mathbf{w} + \mathbf{w}^T\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}}{2\sigma^2} + \frac{\mathbf{w}^T\mathbf{w}}{2b^2}$$

The partial derivatives with respect to $\mathbf{w}$ equals 0 when $\mathbf{w}$ equals $\mathbf{w}_{MAP}$. So,

$$\frac{-2\Phi(\mathbf{x})^T\mathbf{y} + 2\Phi(\mathbf{x})^T\Phi(\mathbf{x})\mathbf{w}_{MAP}}{2\sigma^2} + \frac{\mathbf{w}_{MAP}}{b^2} = 0$$

Then,

$$(\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \frac{\sigma^2}{b^2}\mathbb{I}_M)\mathbf{w}_{MAP} = \Phi(\mathbf{x})^T\mathbf{y}$$

$(\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \frac{\sigma^2}{b^2}\mathbb{I}_M)$ is invertible as $\frac{\sigma^2}{b^2} > 0$. So,

$$\mathbf{w}_{MAP} = (\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \frac{\sigma^2}{b^2}\mathbb{I}_M)^{-1}\Phi(\mathbf{x})^T\mathbf{y}$$

Let $\lambda \in \mathbb{R}^+$ such as $\lambda = \frac{\sigma^2}{b^2}$. Then,

$$\mathbf{w}_{MAP} = (\Phi(\mathbf{x})^T\Phi(\mathbf{x}) + \lambda\mathbb{I}_M)^{-1}\Phi(\mathbf{x})^T\mathbf{y}$$

2.1.3 Conclusion

We can observe that:

$$\mathbf{w}_{MAP} = \mathbf{w}_{ridge}$$

Thus, the linear regression with the presented regularized least squares loss function is equivalent to the MAP estimate for $\mathbf{w}$ with the factorized Gaussian likelihood $y_i \sim \mathcal{N}(\mathbf{w}^T\phi(\mathbf{x_i}), \sigma^2)$ and the following prior for $\mathbf{w}$:
$\mathbf{w} \sim \mathcal{N}(0, b^2)$ with $\lambda = \frac{\sigma^2}{b^2}$.

2.1.4 Intuition of the loss function

In the ridge regression, we add a term to the loss function: $\lambda\mathbf{w}^T\mathbf{w}$.
By adding this term, we force our model to find a solution for w with a low norm. In other words, we forbid the model to find a complex solution for w, such complex solutions would lead to over-fitting as we previously saw.
If λ is too low, the term $\lambda\mathbf{w}^T\mathbf{w}$ won't be high enough to avoid over-fitting. If λ is to high, the model will only accept extremely simple solutions for w which might lead to under-fitting. However, a balanced value of lambda will lead to a well-fitted regularized solution.

2.2 Question b

Firstly, let's observe the impact of the parameter λ on the training and testing errors thanks to a leave-one-out cross-validation.

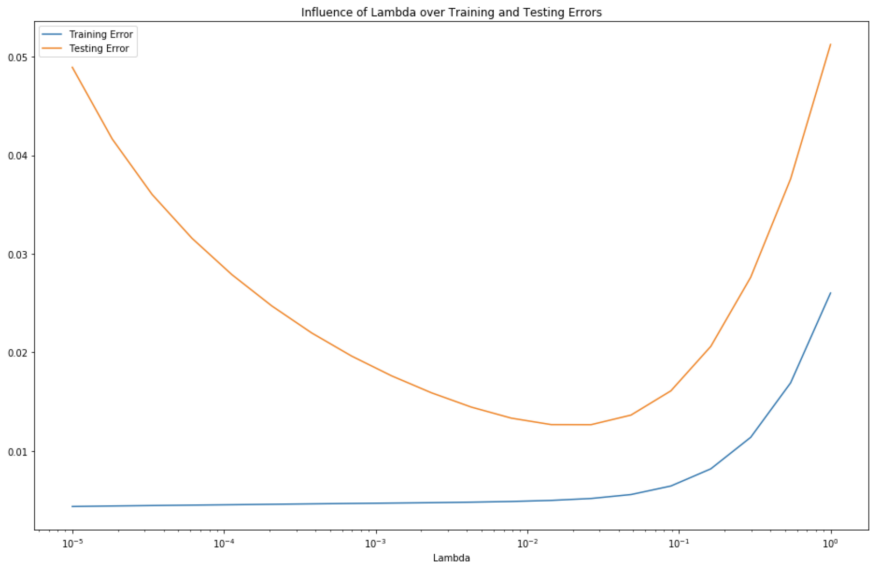

Figure 4: Impact of Lambda over training and testing errors

On Figure 4, we observe that a high value for lambda (10 for example) leads to under-fitting. We observe that a low value for lambda (10_{-15} for example) leads to over-fitting. The optimal value seem to be close to $\lambda = 0.02$. Let's plot these three examples on Figure 5.

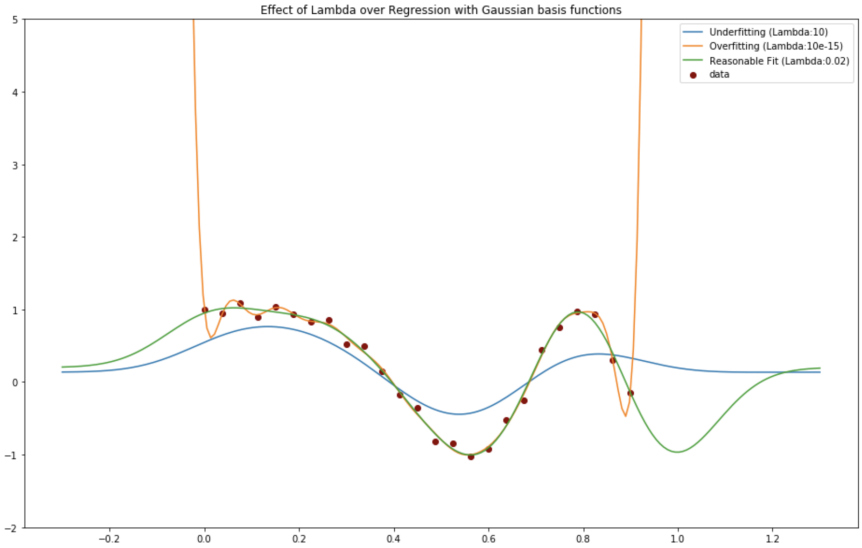

Figure 5: Ridge Regressions for 3 values of Lambda