Kernel Ridge Regression

Machine Learning for Data Science Homework #3

Vid Stropnik; 63200434

ABSTRACT

For the third homework in the Machine Learning for Data Science Class, I worked with kernels. More specifically, I was to comprehend their practical application for using already known linear methods (ie Regression) on non-linearly seperable data.

In this homework, I present my own implementation of Kernel Ridge Regression and a quick exploration into suitable parameters of it.

1 KERNELS AND PROCEDURE

In the homework, we implemented two kernels;

- **Polynomial kernel**; $\kappa(x, x') = (1 + xx')^M$
- **RBF Kernel**; $\kappa(x, x') = \exp\left(\frac{\|x-x'\|^2}{2\sigma^2}\right)$,

And used them to implement the Kernel Ridge Regression, with the prediction formula

$$\hat{y} = k(x')(K + \lambda I_n)^{-1} y,$$

where $k(x')$ and K are comprised of kernels, as seen in the lecture notes. Consequently, no mappings of the input data were needed to achieve the results of our Regression.

2 ROLE OF PARAMETERS AND PROOF-OF-CONCEPT

We were to show both of the kernels 'at work' on the 1-dimensional Sine dataset. A good fit to the input data is shown in Figure 1. This is achieved by first preprocessing the data using `sklearn`'s `StandardScaler`. By tweaking the hyperparameters, we can observe that the quality of the fit of the Polynomial kernel decreases with the size of M, as the plot is not complex enough (doesn't form enough hills and valleys) to consider all of the datapoints. Conversely, with too high a value of M, jagged edges start appearing. The final decision here were the values $\mathbf{M = 10}, \lambda = \mathbf{1}$.

When considering the RBF kernel, anything under the value of 1, with the present value of λ didn't achieve the plot we were after (forming highly frequency curves), so a value slightly over 1 was ideal. With the value of λ, we could then control the phase of the curve. Our best fit was achieved with $\sigma = \mathbf{1.1}, \lambda = \mathbf{1} * \mathbf{10}^{-5}$.

3 HOUSING DATASET & PARAMETER SELECTION

Finally, we were to apply the Kernel Ridge regression to a practical dataset and select the best parameters $\sigma$and$M$ for different values of the regularization parameter λ. The results of the conducted experiments can be seen in Figure 2, with one line representing a constant regularization parameter and the other the best selected regularization parameter for each considered value of $\sigma$and$M$ - in the range between 0.05 and 10, for a step size of 0.05.

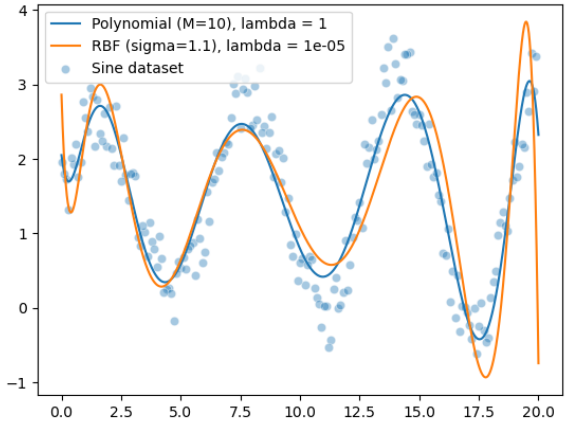

Figure 1: Plot of the two kernels' fits on a simple Sine dataset. The prediction data is generated on normalized data from the range $[0, 20]$.

M	λ	σ	λ
1	4.25	1.5	1.8
2	4.4	2	4.7
3	0.05	2.5	1.4
4	0.05	3	9.95
5	9.95	3.5	5.15
6	9.4	4	3.1
7	4.85	5.5	1.55
8	4.4	7	0.8
9	8.85	8.5	0.5
10	9.95	9.5	0.35

Table 1: The optimal values of the learning rate λ for different values of the tested hyper parameters, considered in the dark green plot in the Figure 2.

The selection of best value of λ was carried out using 5-fold cross validation (on only 80% of all data - the training data for the final plot [1], with most of the results visible from Table 1. The considered hyper-parameters M were in the range of $[1, 10]$, while the considered σ values ranged between 1.5 and 9.5. In initial testing, smaller values of σ were also analyzed, but are not incldued here for the sake of presentation, as the RMSE for a $\sigma < 1$ ended up being much larger (and made the second subplot of Figure 2 less clear. We can observe from said figure, some of the intuition already

[1] I emphasize this because of an initial error made in a previous submission.

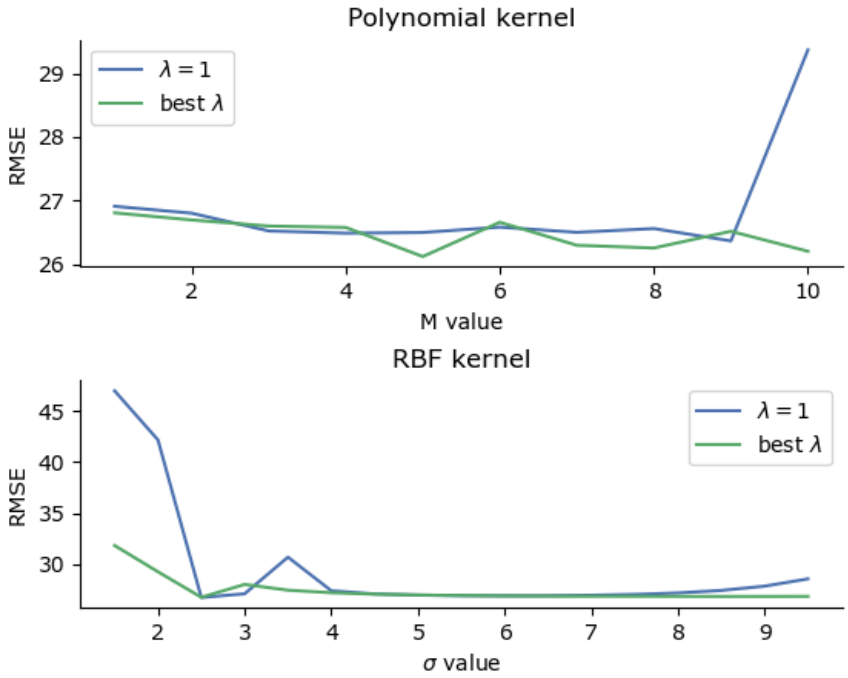

Figure 2: RMSE values for prediction conducted with different values of M and σ. The values of λ, considered in the dark green plots are accounted for in Table 1 for most of the considered hyper parameters.

presented in the previous section - The accuracy of the fit does tend to get better with an increasing value of M, until we begin to overfit. We can notice, however, that an adaptable (notice the stark rise in the value of λ in Table 1 as M gets large!) learning rate is able to circumvent even this shortcoming.

The second subplot also confirms our hypothesis about small values of σ, which stabilizes much faster wth an adaptable learning rate, but nevertheless reaches a steady RMSE plateu soon, even with the constant learning rate $\lambda = 1$.

4 CONCLUSION

In this homework, we showed that complex data may be modeled using simpler (often considered "linear") techniques using the all-powerful Kernel trick. We studied the two presented Mercer kernels and reached an understanding of their hyper-parameters. Seeing as the mapping of data into complex spaces is time and space consuming (or, in the example of infinite spaces, also sometimes infeasible!), I see the kernel trick is a powerful tool, welcome in this data scientist's ever growing arsenal.