

# Machine Learning For Data Science I,

## Homework 05

Maj Gabersček, 27212075

May 2023

## 1 Problem

The problem of this homework was to implement two kernel classes (the polynomial kernel and the RBF kernel) and two regression methods (kernelized ridge regression and support vector regression).

## 2 Implementation

### 2.1 RBF kernel

This implementation was nothing really special. We set the sigma parameter, when initializing the object. When calling it, we check for  $\mathbf{x}$  and  $\mathbf{y}$  dimension and return kernel values.

### 2.2 Polynomial kernel

Again, we set the dimension to the polynomial kernel, when initializing it. Later, when calling it, we firstly check the dimension of the inputs and return correct kernel values.

I also had to add the  $\gamma$  parameter for scaling. Otherwise, polynomial kernels could not be combined with SVR (because of some trouble with `cvxopt.solvers.qp` optimization).

### 2.3 Kernelized ridge regression

When fitting the kernelized ridge regression, we first calculate the kernel matrix,  $K$ . Regarding the closed form solution we set weights as

$$w = (K + \lambda I)^{-1}y.$$

When predicting, we calculate  $k'x = \text{kernel}(X_{\text{train}}, X_{\text{predict}})$  and dot multiply it with weight matrix.

### 2.4 Support vector regression

When fitting SVR, the main goal is to understand, that we must convert problem in Equation 10 of [1] to the QP problem, where we are minimizing  $\frac{1}{2}x^T Px + q^T x$ , subject to  $Gx \leq h$  and  $Ax = b$ . Most challenging was finding a matrix  $P$ , as instructions stated that  $x$  had to be of form  $x = [\alpha_1, \alpha_1^*, \dots, \alpha_n, \alpha_n^*]$ . So after some thoughts, I found out, that:  $P = p_{ij}$ , where  $p_{ij}$  is a block of size  $2 \times 2$ :

$$p_{ij} = \begin{bmatrix} \langle x_i, x_j \rangle & -\langle x_i, x_j \rangle \\ -\langle x_i, x_j \rangle & \langle x_i, x_j \rangle \end{bmatrix}$$

Other matrices were not that hard to find. After parameter optimization with the help of `cvxopt.solvers.qp` function, we set alphas and support vectors to the model.

## 3 The sine dataset

Here, the main goal was to find some parameters, for which the method/kernel would perform relatively good. I plotted performance of models (in regard to MSE on training set). I did not include the plots in the report, but they can be found in plots folder on my [Github repository](#). Of course green color means better performance and red color worse.

So by finding some good parameters, I applied models to the sine dataset, marked the fit, the data and (if the model is SVR) the support vectors (Figures 1, 2, 3, 4). Parameters, that were used for fitting, are printed in the title of every plot.

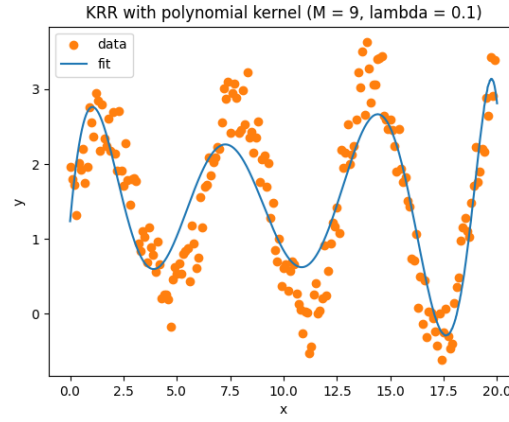


Figure 1: The fit and data for kernelized ridge regression with polynomial kernel

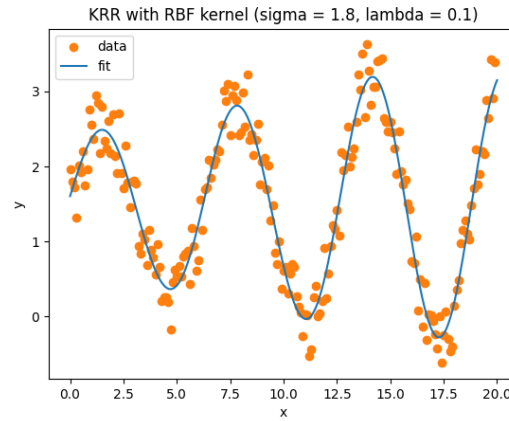


Figure 2: The fit and data for kernelized ridge regression with RBF kernel

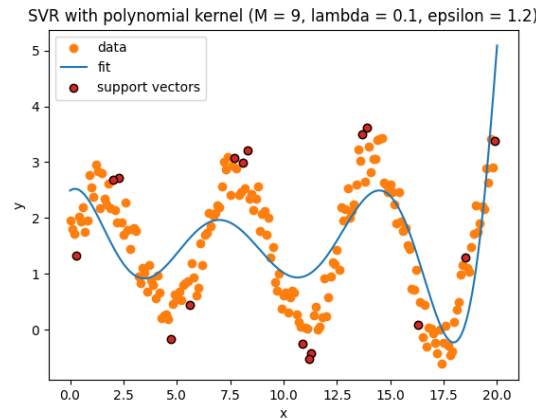


Figure 3: The fit and data for SVR with polynomial kernel

For SVM, I also produced a sparse solution. That is done by checking the weights (more specifically the weight difference  $\alpha_i - \alpha_i^*$ ). If weights are very small, we can exclude the corresponding vector from

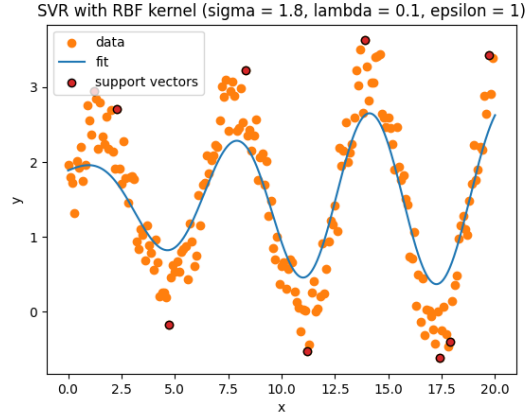


Figure 4: The fit and data for SVR with RBF kernel

our prediction.

## 4 The housing dataset

Next, I applied both regression methods with combination of both kernels to the housing dataset. I plotted MSE in regard to  $M$  (polynomial kernel) and  $\sigma$  (for the RBF kernel). As I didn't really know what values to compare for  $\sigma$ , I decided to use the logarithmic scale.

One plot was for  $\lambda = 1$  and the other plot was for optimal  $\lambda$ , set with internal cross validation. By doing this, we should reduce the risk of overfitting. However, we can still see, that models with  $\lambda = 1$  sometimes overperform models with optimal  $\lambda$ , when calculating the MSE on a separating testing set. Plots for MSE can be seen in Figure 5.

For support vector regression, I also checked how number of vectors included affect the MSE. I plotted how MSE changes in regard to number of support vectors for both kernels and all parameters. I filtered out vectors one by one, first those, whose weight was the smallest and so on. I did not include all the visualizations in my report. Again, they are included in my [Github repository](#). Just to paint a picture, I am including two samples in Figures 6 and 7. We can see, that it would be safe to take 70 vectors (in first example) and around 90 vectors (in the second example).

## 5 Conclusions

As we can see in the sine dataset, the kernelized ridge regression was able to predict the fit better (more accurately). However, when computing the model on the housing dataset, SVR proved to be more stable. For example, the kernelized ridge regression in combination with polynomial kernel generated MSEs of great magnitude.

Regarding my preferences, I preferred the kernelized ridge regression, simply for the fact, that I did not need to use additional libraries and packages for optimization.

## References

- [1] Alex J Smola and Bernhard Schölkopf. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222.

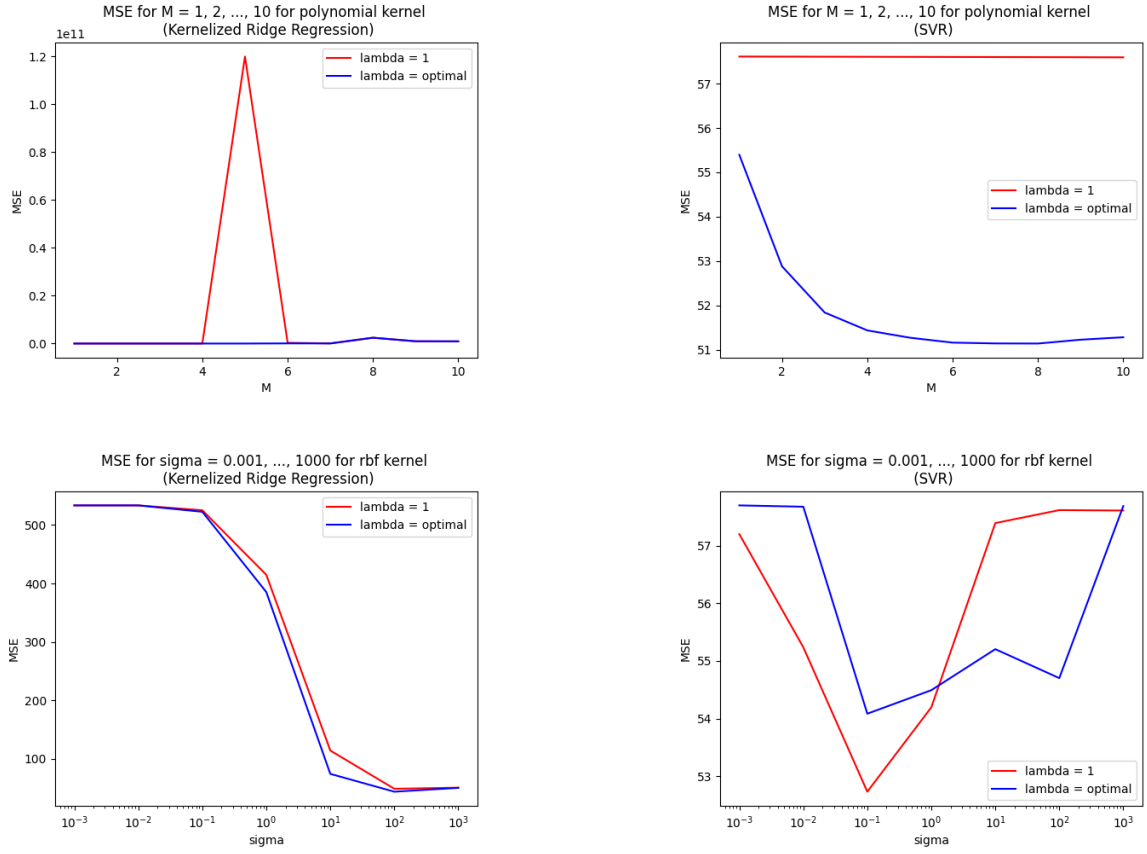


Figure 5: MSE on the test set in regard to parameters

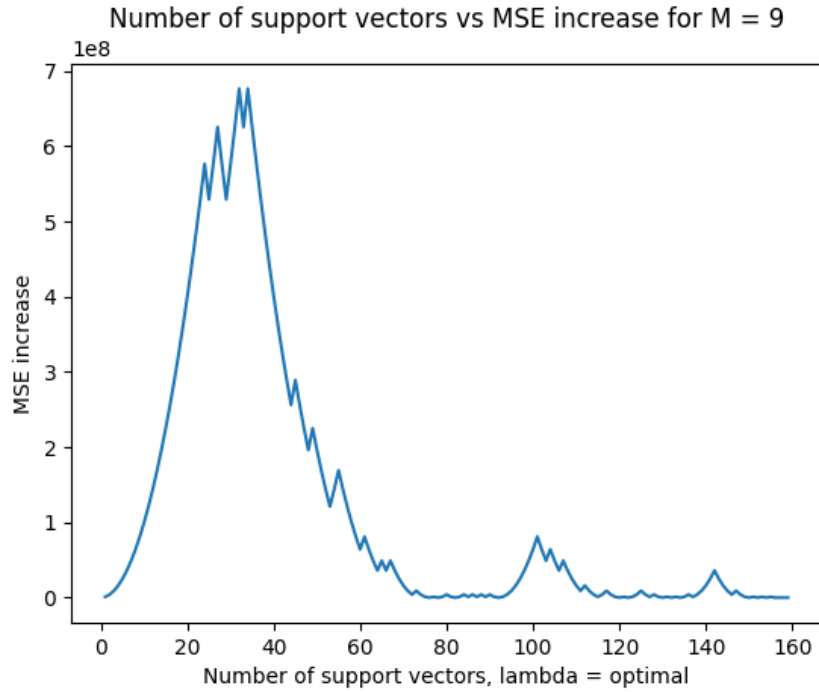


Figure 6: Percentual MSE increase in regard to support vector number, polynomial kernel with  $M = 9$  and optimal  $\lambda$

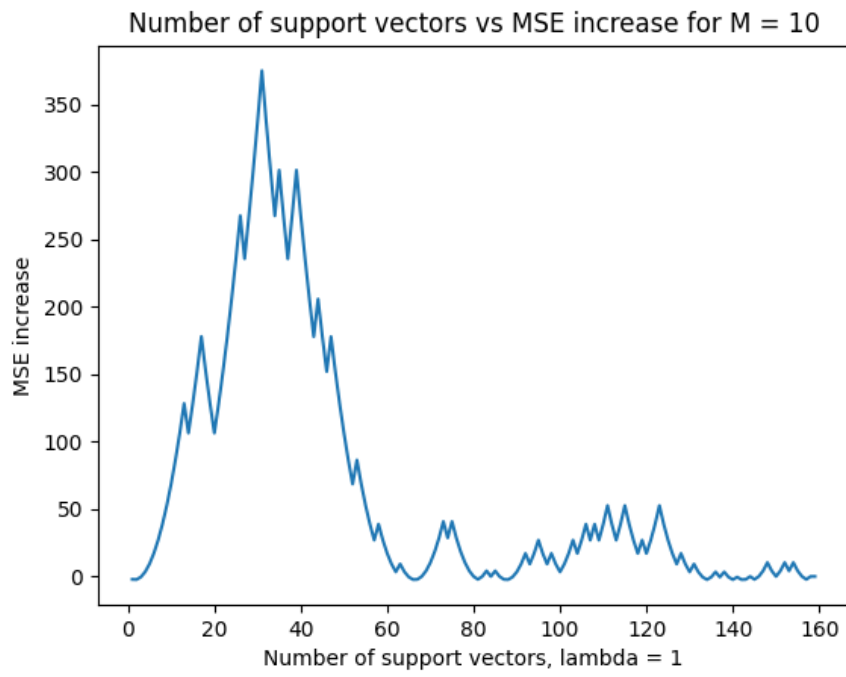


Figure 7: Percentual MSE increase in regard to support vector number, RBF kernel with  $\sigma = 10$  and optimal  $\lambda$