# Support Vector Regression

Urša Zrimšek

## Introduction

This is the report of fourth homework for class Machine Learning for Data Science 1, in which we implemented support vector regression.

We used two kernels:

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

The models are applied to two datasets: one dimensional `sine.csv` dataset, that is composed of 200 rows with $x$ and $y$ values, and `housing2r.csv` dataset, that has 200 rows with 5 independent and one dependent column. We analyzed this method and compared it to kernelized ridge regression, implemented in the previous homework.

## Sine dataset

On Figure 1 we can see the fit of model using support vector regression with polynomial and RBF kernels. To get this fit we chose regularization parameter $\lambda = 0.1$, degree of polynomial kernel $M = 15$ and width of RBF kernel $\sigma = 0.2$ - because of the same reasons as in HW3. We chose $\epsilon = 0.5$, because with bigger $\epsilon$ the fit was worse, and with smaller ones our solution was not sparse anymore - almost all points were selected as support vectors.

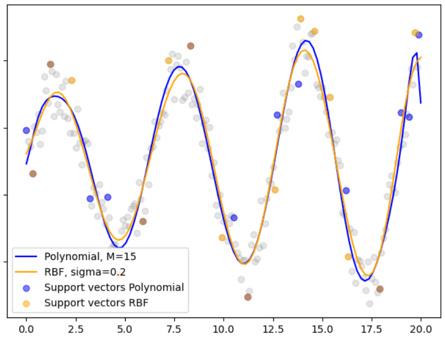

Figure 1. **Sine data, the fit of support vector regression with polynomial and RBF kernel and the support vectors from both methods**. Beware that some support vectors of both methods are the same.

## Housing dataset

On housing dataset we took the first 80% as the training set, and last 20% as test set. We compared RMSE for both kernels with different parameter values - $M$ and $\sigma$. For each value of those two parameters, we did a 5- fold cross validation on the training data, to set the best value of regularization parameter $\lambda$. Then we plotted RMSE versus the parameter values, for models with $\lambda = 1$ and with $\lambda$ chosen with cross validation, where we were choosing from $\lambda \in \{0.1, 0.25, 0.5, 1, 2.5, 5, 10\}$. We calculated RMSE for $M \in \{1, \dots 10\}$ and for 100 $\sigma$-s in between 0.01 and 20.

We set $\epsilon$ to 8, as this value minimized the RMSE together with low number of support vectors in both methods.

The errors and the number of support vectors can be seen on Figures 2 and 3.

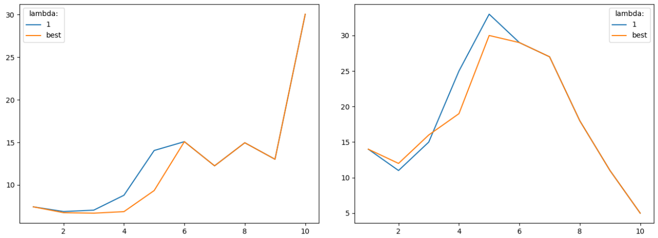

Figure 2. **RMSE (left) and number of support vectors (right) versus the degree of Polynomial kernel.** We can observe that the error is large at bigger degree kernel, so we can conclude that this data is simple, and we overfit it quickly. The lowest error is achieved at $M = 2$ for constant $\lambda$ and a bit lower for $M = 3$ for best $\lambda$. On the best results the number of vectors is sparse - between 10 and 15. If we compare this results with the previous homework, we can see that they are much better at big degrees, but the smallest RMSE is achieved with ridge regression - 6.5 at $M = 2$. Here the smallest error is 6.7.

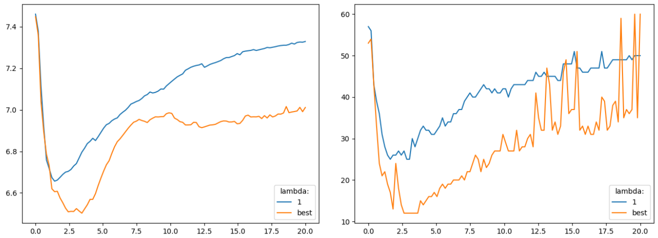

Figure 3. **RMSE (left) and number of support vectors (right) versus the width ($\sigma$) of RBF kernel.** The error is large at the smallest $\sigma$s, that is, as we saw in the previous homework, the consequence of overfitting. But here the values again raise with bigger values. It could be the consequence of too much generalization there. We reach the smallest error at $\sigma = 3.4$ and $\lambda = 0.1$, and with $\lambda = 1$, the best value for $\sigma$ is 1.4. Again, the results are much better with support vector regression where we select wrong parameter values (at the small sigmas), but with optimal parameters, the error is quite similar. At the lowest error, the number of support vectors is similar as with polynomial.

We can again observe the values of best $\lambda$. For RBF, we looked at each tenth value, and they settled at 0.1. But for polynomial kernel they are equal to [1, 0.25, 10, 10, 10, 5, 0.1, 0.1, 1, 0.25]. We can see that they lower towards the end, which could mean that our first hypothesis about overfitting at bigger degrees is not correct. If it would hold, the results should be better at bigger values of regularization parameter. If we look at the number of support vectors for the big values, it looks like the method compensates well for it with lower number of support vectors, and that could be the reason for better performance with these parameters.

## Conclusion

Compared to kernelized ridge regression, this method is much better when we don't choose optimal parameters. When we do, they reach similar results. Here there are more parameters to set, but we have more space to make mistakes, and still reach good result, so this should be the prefered algorithm, specially when we don't have much insight into the data we are modeling.