

Kernels and Support Vector methods

Aljaž Konec

I. INTRODUCTION

In this report we showcase implementation of Support Vector Methods for regression tasks. Specifically, we focus on Support Vector Regression (SVR) and Kernelized Ridge Regression (KRR). For both methods we use the kernel trick with the Polynomial and RBF Kernels to compute encodings.

II. IMPLEMENTATION: KERNELIZED RIDGE REGRESSION

From normal Ridge regression, we derive the Kernelized Ridge Regression (KRR) by replacing the dot product of X and X' with the kernel matrix K . This gives a closed form solution for the weights α :

$$\alpha = (K + \lambda \cdot I)^{-1} \cdot y$$

where λ is the regularization parameter and y is the target vector.

III. IMPLEMENTATION: SUPPORT VECTOR REGRESSION

To find support vectors for SVR, we find the Lagrange multipliers α and α^* that maximize the dual problem:

$$-\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*)$$

subject to

$$\sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \quad \text{and} \quad \alpha_i, \alpha_i^* \in [0, C]$$

Similarly to KRR, this gives us the weights $\alpha_weight = \alpha - \alpha^*$.

IV. TESTING ON SYNTHETIC SINE DATA

First we showcase the performance on synthetic sine data where we trained both SVR and KRR with Polynomial and RBF kernels on the entire dataset. To find good performing hyperparameters, we used grid search for each of the methods/kernels and used best performing hyperparameters. Figure 1 shows the results of predictions using SVR and KRR with both Polynomial and RBF kernels. Using RBF kernel for both methods approximates the sine data very well, while Polynomial kernels result in a poor fit. The sine curve can be approximated with a Taylor expansion, but the Polynomial kernel of degree 4 is not enough to capture this expansion. The number of support vectors for SVR with RBF kernel is 23, while for SVR with Polynomial kernel is 6.

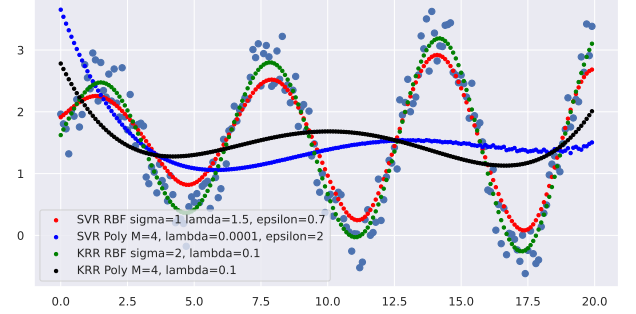


Figure 1: Predictions of sine data with SVR and KRR both with Polynomial and RBF kernels.

For Polynomial kernels we also tried to increase the degree but that resulted in the program giving an error. We gather that this is due to numerical instability when using higher degrees, as the kernel matrix becomes ill-conditioned.

V. PREDICTIONS ON HOUSING DATA

Similarly to the previous homework, we used the housing data to test the performance of SVR and KRR. For better performance, we first normalized the data. Figure 2 shows the root mean squared error (RMSE) of the predictions for both kernels with different kernel parameters. The first column represents the predictions using KRR, while the second column represents the predictions using SVR. We show how regularization parameter λ affects the performance of both methods. The blue lines represent the RMSE with fixed $\lambda = 1$, while the orange represent the RMSE where we use cross-validation to find the best λ for each kernel parameter. To find the best λ , we used GridSearchCV from sklearn with 5-fold cross-validation on the training data. The tested λ -values were $[0.0001, 0.001, 0.01, 1, 2, 3, 4, 5, 10]$. The parameter $\epsilon = 8$ gave the best results for SVR and was kept fixed. For SVR we also show the number of support vectors used for each kernel parameter.

Choosing a σ that is smaller than 1 resulted in very poor performance for all tested models and lambdas. This is due to the fact that the RBF kernel is not able to capture the complexity of the data with such a small σ as the resulting gaussian is too narrow. Choosing higher values for σ did not change the performance of KRR significantly, while for SVR it resulted in a better performance. SVR with dynamic cross-validation lambda performed better quite better than SVR with fixed lambda and achieving its minimum at $\sigma = 3$. The number of support vectors for SVR with RBF kernel also decreases if using dynamic cross-validation lambda. This is due to the fact that the model is able to generalize better and does not need as many support vectors to capture the complexity of the data.

For Polynomial kernels we can see that increasing the degree to $M = 5$ results in worse performance for both models. This is due to the fact that the model becomes too complex and is not able to generalize well. The number of support vectors subsequently also increases with the degree of the polynomial

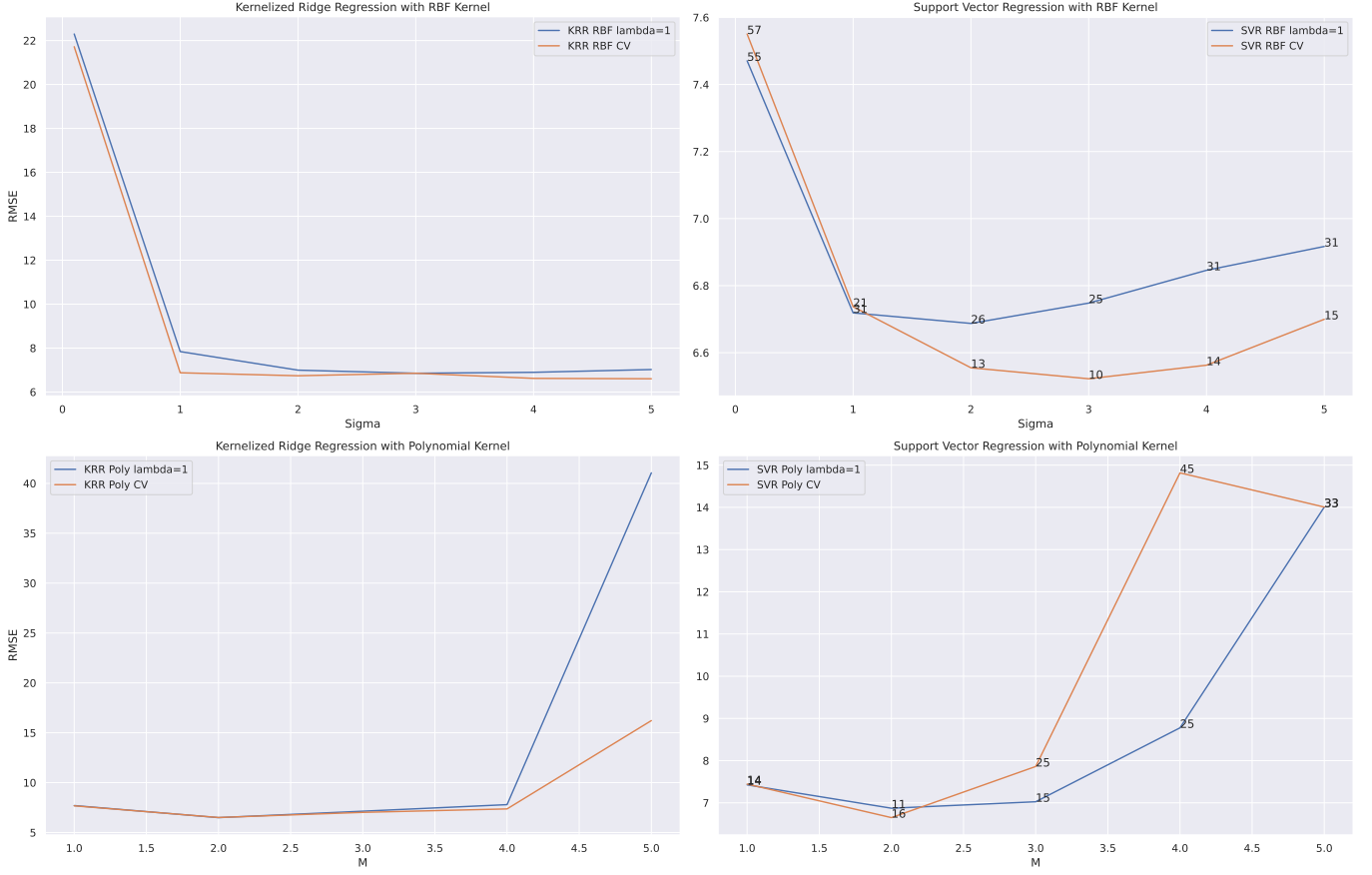


Figure 2: Predictions of housing data with SVR and KRR both with Polynomial and RBF kernels.

kernel. Performing dynamic cross-validation lambda resulted in similar performance for KRR with an improvement in performance for $M = 5$. Interestingly, using dynamic cross-validation lambda for SVR resulted in worse performance for M larger than 3. Since this is a small dataset, we conclude that this is due to the lambda being overfitted to the training data.

Overall for the right choice of hyperparameters, all models and kernels achieve similar performance with RMSE between 6 and 7. In general, using the RBF kernel resulted in slightly better and more stable performance than the Polynomial kernel. Using dynamic cross-validation lambda resulted in better performance but might be difficult to deploy to larger datasets.

VI. CONCLUSION

In this report we saw that both methods are able to capture the properties of the data and achieve similar performance. The main difference between the two methods is that SVR does not have a closed form solution and may sometimes not be a feasible optimization problem. SVR also has a hyperparameter ϵ that needs to be taken into consideration when tuning the model. On the other hand, KRR has a closed form solution but uses the inverse of a potentially very large kernel matrix. Given all of this we would prefer to use SVR with RBF kernel as it offers the best performance and for larger datasets we do not need to compute the inverse of a matrix.