Kernels

Blaž Erzar

Machine Learning For Data Science 1 2023/24
Homework 5 report

Abstract

This report presents the implementation and evaluation of two kernel methods, kernalised ridge regression and support vector regression. Ridge regression can be kernelised using an alternative expression of the closed-form solution, while SVR is fitted by solving an optimisation problem. For both methods, we define two kernels, the polynomial and the RBF kernel. We evaluate combinations of both methods and both kernels on a one-dimensional sine dataset and the housing dataset. We compare mean squared error for different values of kernel and regularisation parameters. In the end, the SVR and RBF kernel perform the best.

1 Introduction

In this homework, we implement two kernelised methods, *kernelised ridge regression* and *support vector regression*. In both methods, we use two kernels, the polynomial and the RBF kernel. We use a Python package `cvxopt`, for solving convex optimisation problems, to solve the optimisation problem defined in [1]. For SVR, we set ε such that the number of support vectors is not too high and we obtain a sparse solution.

Both methods are evaluated on two datasets. First, we fit a small dataset, representing the sine function, and compare how different kernels perform. Lastly, we look at the MSE, and the number of support vectors for SVR, on the test set of the housing dataset. We compare results for $\lambda = 1$ and λ selected using 5-fold cross-validation.

2 Methods

2.1 Kernels

Kernels can be viewed as a similarity measure between two feature vectors. The polynomial kernel of degree M is computed as

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\mathsf{T}\mathbf{x}')^M,$$

and the RBF kernel with bandwidth σ as

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2/(2\sigma^2)).$$

For a matrix of instances, we can efficiently implement both of them using matrix multiplication.

2.2 Kernelised ridge regression

The ridge regression solution can be alternatively expressed as

$$\beta = \mathbf{X}^\mathsf{T}\left(\mathbf{X}\mathbf{X}^\mathsf{T} + \lambda\mathbf{I}\right)^{-1}\mathbf{y}.$$

When doing predictions for instances $\mathbf{X}'$, we compute $\hat{\mathbf{y}} = \mathbf{X}'\beta$, which becomes

$$\hat{\mathbf{y}} = \mathcal{K}(\mathbf{X}', \mathbf{X})(\mathcal{K}(\mathbf{X}, \mathbf{X}) + \lambda\mathbf{I})^{-1}\mathbf{y}$$

using the kernel trick.

2.3 Support vector regression

The optimisation problem we are solving is

$$\max \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases}$$

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

After solving it, we predict using

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + b,$$

where bias b can be obtained from the solver or computed using the solution. Instances for which $\alpha_i - \alpha_i^* \neq 0$ are called the **support vectors**. All other instances do not affect the prediction.

The matrices required by the solver can be efficiently created using Kronecker product and some basic stacking `numpy` operations.

2.4 Data preprocessing

For SVR, we have to normalise all features to the interval $[0, 1]$, otherwise, we get numerical errors in the optimisation package.

3 Results

3.1 Sine dataset

The results of fitting the sine dataset are shown in Figure 2. Both ridge regression and SVR perform

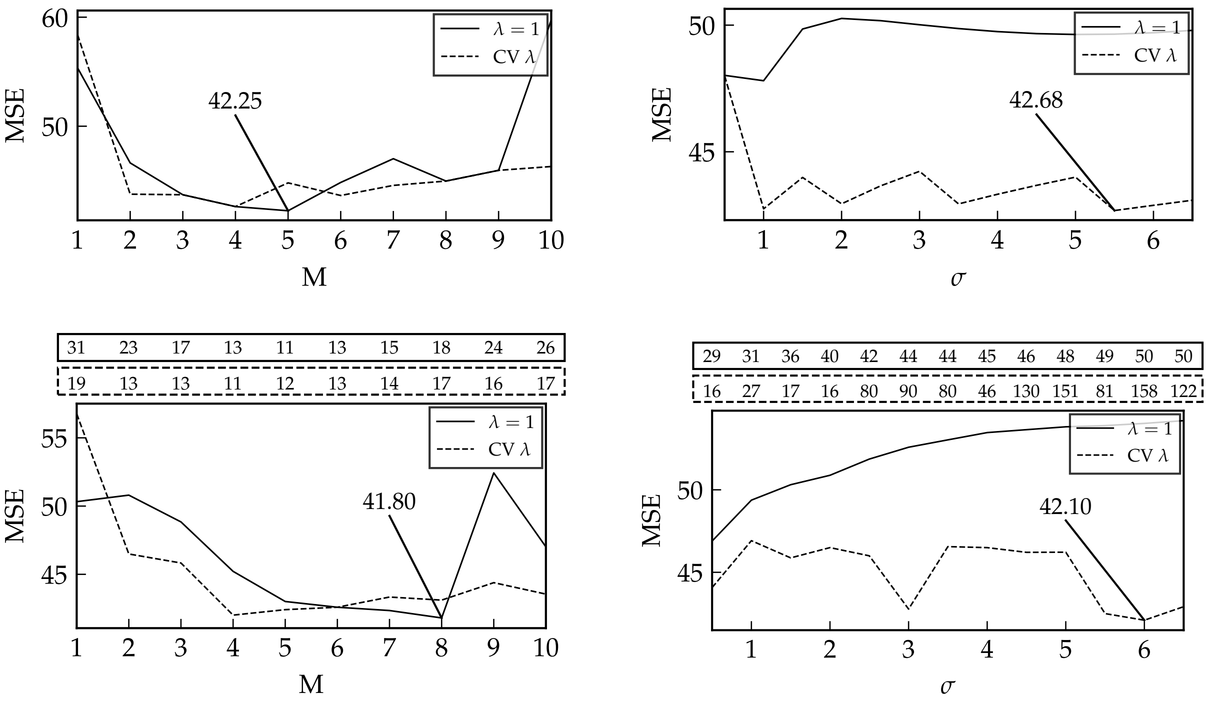

Figure 1: *Mean squared error for different parameters of the polynomial kernel (left) and the RBF kernel (right), with kernelised ridge regression (top) and SVR (bottom). For SVR, we also show the number of support vectors in boxes with border styles corresponding to the plot lines. We used* $\varepsilon = 8$ *for SVR.*

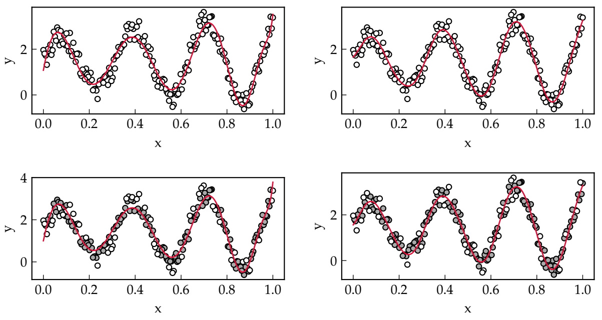

Figure 2: *The sine dataset fit for kernelised ridge regression (top) and SVR (bottom) with polynomial kernel with* $M = 19$, $\lambda = 1\text{e}{-5}$ *(left) and RBF kernel with* $\sigma = 0.1$, $\lambda = 1\text{e}{-3}$ *(right). Support vectors are shown with white dots and other data points in grey.*

similarly for the same kernel parameters. The difference is that the SVR generates a sparse solution. For the polynomial kernel, 24.5% of instances are support vectors and only 18% for the RBF kernel. RBF also has a slightly better fit and works better if parameters are not optimal. For the polynomial kernel, the degree has to be sufficiently high.

3.2 Housing dataset

The results on the housing dataset are shown in Figure 1. We use the last 20% of the 200 instances for testing. Using the polynomial kernel, both λ values perform similarly, with $\lambda = 1$ being better at $M = 5$ for the ridge regression and $M = 8$ for the SVR. For the RBF, the regularisation parameter obtained using cross-validation always works better.

For the lower values of σ, we get a similar amount of support vectors as with the polynomial kernel, but for higher values, the number is a lot bigger. As we can see, the SVR obtains lower MSE for both kernels.

4 Discussion

For the datasets we are using, both methods and kernels perform similarly. RBF kernel does seem easier to fine-tune because polynomial degrees can be quite unstable and cause problems for higher values.

Comparing the learning algorithm, the SVR does take longer to compute, but the MSE is lower and the obtained solution is sparse. This could be important for larger datasets, so we do not have to keep the whole **X** in memory, but just keep the support vectors.

References

[1] Alex Smola and Bernhard Scholkopf. “A tutorial on support vector regression”. In: *Statistics and Computing* 14 (2004), pp. 199–222. URL: `https://api.semanticscholar.org/CorpusID:15475`.