Kernelized ridge regression

Žiga Trojer - 63200440

1. Introduction

We implemented kernelized ridge regression with two kernels:

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

We implemented both kernels such that computations are fast and it works for multiple dimensions of input (two vectors, vector - matrix and two matrices).

2. Results

2.1 Sine data

First we tested how well we can fit 1-dimensional data set. We were provided with `sine` data. We tested a lot of parameter values and found out that if we want to fit our data well, we need to scale the data (we used `StandardScaler` from `sklearn.preprocessing`). If we don't, the fit is bad. This is especially evident in fit where we use a polynomial kernel. Figure 1 shows our best fit of `sine` data. For the polynomial kernel, our choice of parameters were: $M = 11$ and $\lambda = 0.001$. We saw that if we chose parameter M too small, our fit is not good - too 'simple' approximation in terms of kernel degree. If we chose too M too large, fit would be bad too - this already happens for $M = 14$. We can see potential problems with this kernel, since we need to change this parameter often, if we would add more data (for example, one more sine wave). For the RBF kernel, parameters were: $\sigma = 0.3$ and $\lambda = 0.001$. Here also, too small σ would over-fit the data and too big σ would not fit the data good enough. We also experimented with regularization parameter λ and noticed that is should not be 0, but something small. For big values of λ, the fit is worse, since it does not cover curves very well. WE also notice that the fit for first curve is a bit shifted to the left side - this is probably due to good fit on the left side.

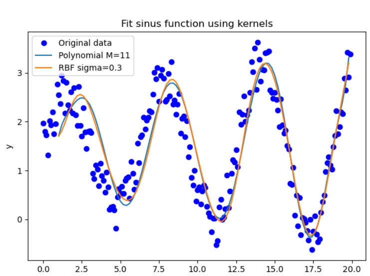

Figure 1. Fitted sine data using both kernels.

2.2 Housing data

We also applied kernelized ridge regression to the second dataset - `housing2r`. We used first 80% of data as a training set and remaining 20% as test set. For both kernels, we tested performance for $\lambda = 1$ and for optimal λ, which was chosen for each value of parameter M and σ. Figure 2 shows RMSE on testing set, using polynomial kernel and different choices of parameters λ - first with $\lambda = 1$ and then with optimal λ for each value of M in range $[1, 10]$, which we got from 5-fold cross-validation. We can observe that RMSE for optimal parameter is not always lower than for $\lambda = 1$, which can happen, since we tested it on test data. Optimal parameters for λ from CV increased with M: for $M = 1$, we got $\lambda = 1$ and with $M = 10$, we got $\lambda = 99$. Too big M increases the RMSE - this means that we should choose lower M for this dataset.

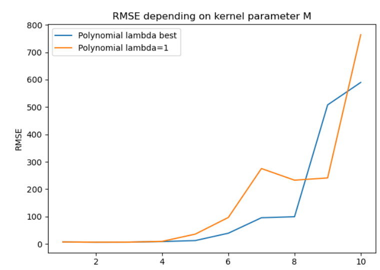

Figure 2. RMSE depending on chosen parameter value M.

Figure 3 shows RMSE using RBF kernel. We searched for optimal value of λ for each σ in range $[0.1, 10]$ with step 0.1. Here also, optimal value of λ increases with σ. With the increase of parameter σ, RMSE descreases. This shows that we over-fit data if σ is too small. We also noticed that σ too big will increase the RMSE (but we did not include big σ's to the graph). For some values of σ, RMSE for optimal choice of λ is higher than for $\lambda = 1$, but it is lower in general. This means that we should always consider both parameters when using those kernels.

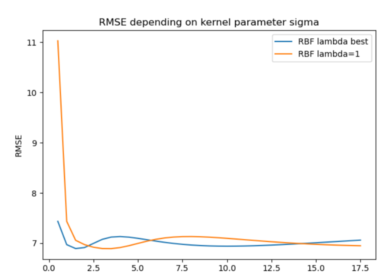

Figure 3. RMSE depending on chosen parameter value σ.