Assignment 2–Solution

EECS 4404/5327, Fall '20

The assignment is due on Monday, October 26, by the end of the day.

In this assignment you will implement linear least squares regression. For the data provided, set up the design matrix (using non-linear basis functions for polynomial curve fitting) and solve for the parameter vector $\mathbf{w}$ as discussed in class. Print out your code and include it in your submission.

Step 1 - load the data
The data is stored in two files, `dataset1_inputs.txt` and `dataset1_outputs.txt` which contain the input values (i.e., values x_i) and the target values (i.e., values t_i) respectively. These files are simple text files which can be loaded with the load function in Matlab/Octave. Plot the outputs as a function of the inputs (ie plot the datapoints, not a curve) and include this plot in your write-up.

Step 2 - ERM
For degrees $W = 1, \ldots 30$, fit a polynomial of degree W to the data using (unregularized) least squares regression. For each learned function, compute the empirical square loss on the data and plot it as a function of W. Include this plot in your report. Given the curve, which value of W do you think would be suitable?

Step 3 - RLM
Repeat the previous step using regularized least squares polynomial regression. Each time train polynomial of degree 30 for regularization parameters λ so that $\ln(\lambda) = -1, -2, \cdots -30$. This time plot (and include) the empirical loss as a function of i. Compare and discuss the two curves you get for ERM and RLM.

Step 4 - cross validation
Implement 10-fold cross validation for ERM. That is, randomly divide that data into 10 chunks of equal size. Then train a model on 9 chunks and test on the 10th that was not used for training. For each model you train, average the 10 test scores you got and plot these again as a function of W. Which value of W do you think would be suitable?

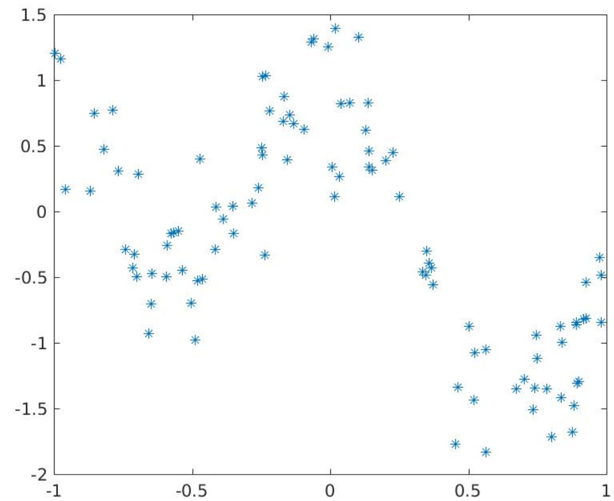
Step 5 - visualization
For the degrees $W = 1, 5, 10, 20, 30$ plot the data along with the ERM learned models. Do the same for models learned with RLM with a fixed regularization parameter $\lambda = 0.0025$ (while varying the degree as for ERM). Discuss the plots. Which degree seems most suitable? What is the effect of adding the regularizer here?

(4 + 4 + 4 + 4 + 4 marks)

Solution

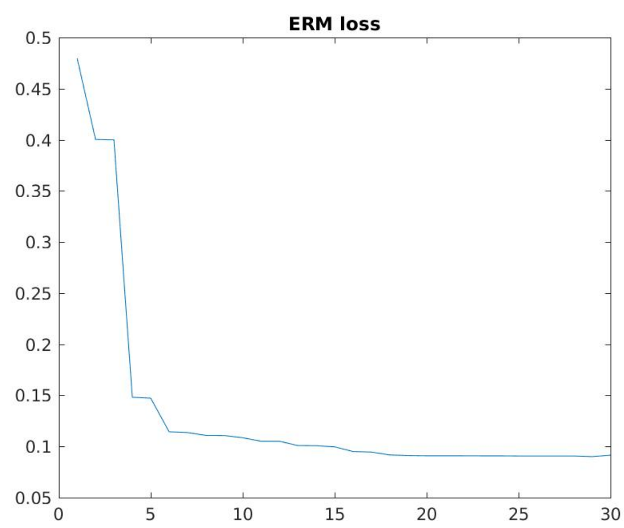
Step 1 - load the data

Here you just load the inputs and outputs and then generate the following plot:



Step 2 - ERM

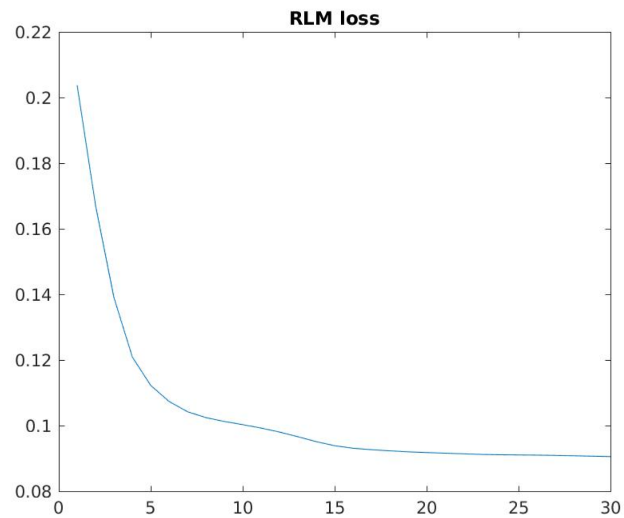
Here you fit a polynomial of various degrees to the data, and compute the training loss as a function of the degree of the fitted polynomial. As expected, the loss decreases with increasing the degree (as we fit potentially more complex functions as the degree increases):



At this point, it seems that a degree of 6 would be suitable. The loss decreases sharply up to degree 6, then continues to decrease only mildly. This indicates that, for degrees larger than 6, we would expect overfitting to occur.

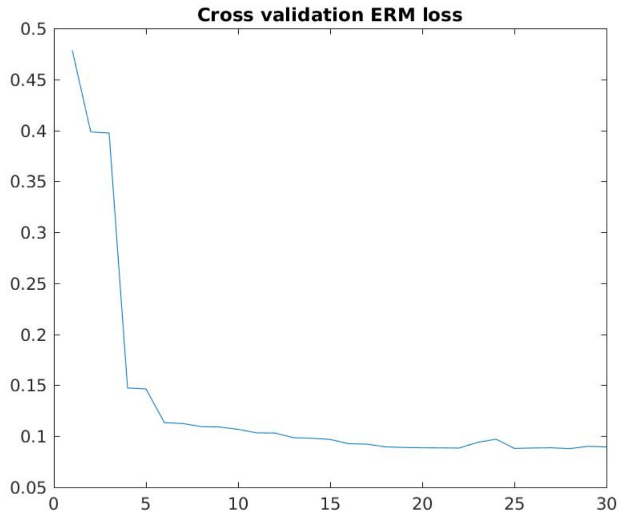
Step 3 - RLM

We repeat the same steps but, now fix the degree of the polynomial to 30 and vary the regularization parameter. Plotting the loss as a function of $i = \ln(1/\lambda)$ yields a very similar picture:



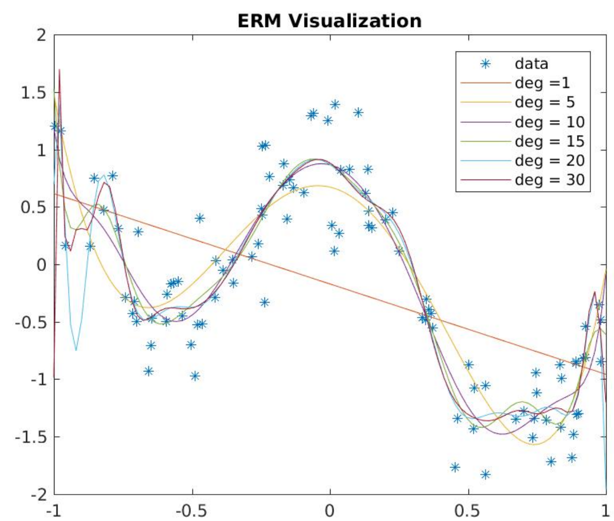
Step 4 - cross validation

Here, we first randomly permute the data. Then we divide it into 10 equal sized parts. Now, for every degree from 1 to 30, we choose one of the 10 parts as testset, fit a polynomial to the other 9 (using the ERM method), and then compute the testloss on the chosen testset. That is, for every degree, we fit a function 10 times, compute the test loss, and then average these 10 testlosses. Plotting these averages as a function of the degree yields:

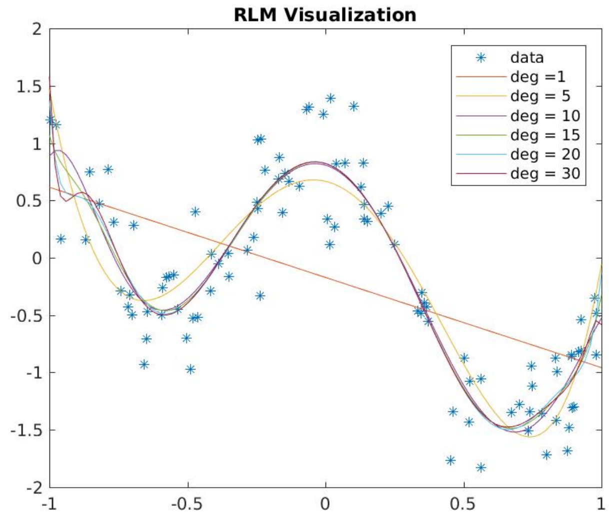


Step 5 - visualization

Plotting polynomials of various degrees trained without regularizing together with the data:



Plotting polynomials of various degrees trained with a regularizer using $\lambda = 0.0025$ together with the data:



We observe how the fitted function follows the trends in the data increasingly with higher degrees. We also see (particularly at the ends), when training without a regularizer, that for degrees higher than 5, the function starts to fit the noise (an indication of overfitting). Adding a regularizer with a small regularization parameter prevents this phenomenon. Out of the chosen degrees for visualization, degree 5 seems most suitable; the polynomials of higher degrees show signs of overfitting.