FMAN45 Machine Learning Assignment 1

Filip Cederquist

March 2020

Penalized regression via the lasso

Using linear regression it is possible to estimate a weight variable $w \in \mathbb{R}^M$ given a linear relationship to the data points $t \in \mathbb{R}^N$ with the regression matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$. In LASSO regression, we add a penalty term to the weights with the objective of reducing overfitting and keeping the model simple. LASSO regression is the same as least square except with this added term. Therefore we want to solve the minimization problem:

$$\underset{\mathbf{w}}{minimize} \quad \frac{1}{2}||\mathbf{t} - \mathbf{X}\mathbf{w}||_2^2 + \lambda|\mathbf{w}|_1 \tag{1}$$

where $\lambda \in \mathbb{R}$ is a hyper parameter to be set beforehand. For the general regression matrix there exists no closed form solution. Instead we choose to look at the coordinate wise update which has a closed form solution.

$$\underset{w_i}{minimize} \quad \frac{1}{2}||\mathbf{r}_i - \mathbf{x}_i w_i||_2^2 + \lambda|w_i|_1 \tag{2}$$

where x_i is the i'th vector of the regression matrix and

$$r_i = t - \sum_{l \neq i} \mathbf{x}_l w_l$$

For the i'th coordinate we have:

$$\hat{w}_i^{(j)} = \begin{cases} \frac{\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^T \mathbf{x}_i |\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}|} \left(|\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}| - \lambda \right) & , \quad |\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}| > \lambda \\ 0 & , \quad |\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}| < \lambda \end{cases} \tag{3}$$

where

$$\mathbf{r}_i^{(j-1)} = \mathbf{t} - \sum_{l<i} \mathbf{x}_l \hat{w}_l^{(j)} - \sum_{l>i} \mathbf{x}_l \hat{w}_l^{(j-1)} \tag{4}$$

The coordinate wise update solves the problem for each coordinate by fixing the others during the update and iterating over the vector.

Task 1

Verify the first line of (3) by solving (2) for $\hat{w}_i \neq 0$.

Solution

$$f(w_i) = \frac{1}{2}||\mathbf{r}_i - \mathbf{x}_i w_i||_2^2 + \lambda|w_i| \tag{5}$$

$$\implies f(w_i) = \frac{1}{2}\mathbf{r}_i^{T(j-1)}\mathbf{r}_i^{(j-1)} - \mathbf{r}_i^{T(j-1)}\mathbf{x}_i w_i + \frac{1}{2}(\mathbf{x}_i w_i)^T \mathbf{x}_i w_i + \lambda|w_i| \tag{6}$$

$$\tag{7}$$

First term is not dependent on w_i so we create a new function $f_2(w_i)$ without this term and minimize that instead.

$$f_2(w_i) = \mathbf{r}_i^{T(j-1)}\mathbf{x}_i w_i + \frac{1}{2}(\mathbf{x}_i w_i)^T \mathbf{x}_i w_i + \lambda|w_i| \tag{8}$$

$$\tag{9}$$

Solve $f_2^{'}(w_i) = 0$. This leads to:

$$\hat{w}_i^{(j)} = \frac{\mathbf{x}_i^T \mathbf{r}_i^{(j-1)} - \lambda \cdot \frac{\hat{w}_i^{(j-1)}}{|\hat{w}_i^{(j-1)}|}}{\mathbf{x}_i^T \mathbf{x}_i}, \qquad \frac{\hat{w}_i^{(j-1)}}{|\hat{w}_i^{(j-1)}|} = \frac{x_i^T r_i^{(j-1)}}{|x_i^T r_i^{(j-1)}|} \quad if \quad |x_i^T r_i^{(j-1)}| > \lambda \tag{10}$$

The sign of w_i is as above due to the denominator of the first term being strictly positive so it depends on the sign of $x_i^T r_i^{(j-1)}$. By insertion and rearanging we arrive at:

$$\hat{w}_i^{(j)} = \frac{x_i^T r_i^{(j-1)}\Big(|x_i^T r_i^{(j-1)}| - \lambda\Big)}{x_i^T x_i |x_i^T r_i^{(j-1)}|} \tag{11}$$

Task 2

Given the orthogonal regression matrix, show that the coordinate descent solver in (3) will converge in at most 1 full pass over the coordinates in $\mathbf{w}$, i.e., show that: $\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = 0, \forall i$.

Solution

First of, since $X^T X = I$ we know that $x_i^T x_i = 1$ and that $x_i^T x_j = 0$ for $\forall i \neq j$. Hence by inserting (4) into (11) all terms except $\mathbf{t}$ in (4) will become 0. This leads to:

$$\hat{w}_i^{(j)} = \frac{\mathbf{x}_i^T \mathbf{t}\Big(|\mathbf{x}_i^T \mathbf{t}| - \lambda\Big)}{\mathbf{x}_i^T \mathbf{x}_i |\mathbf{x}_i^T \mathbf{t}|} \tag{12}$$

Hence, the estimate does not depend on previous estimates $w_i^{(j)}$ and will converge after at most 1 full pass.

Task 3

When $\mathbf{w}$ is estimated from data using the LASSO, a bias will be induced into it, i.e, $E(\hat{w} - w^*) \neq 0$. Assume that the data $\mathbf{t}$ is truly a noisy (stochastic) example from the hypothesis space defined by the regression matrix, that is

$$\mathbf{t} = \mathbf{X}\mathbf{w} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(\mathbf{0}_N, \sigma \mathbf{I}_n). \tag{13}$$

Show that the LASSO estimate's bias, for an orthogonal regression matrix and data generated by (13), is given by (14) when $\sigma \to 0$, and discuss how this result relates to the method's acronym, LASSO.

$$\lim_{\sigma \to 0} E\Big(\hat{w}_i^{(1)} - w_i^*\Big) = \begin{cases} -\lambda & , \quad w_i^* > \lambda \\ -w_i^* & , \quad |w_i^*| \leq \lambda \quad \forall i \\ \lambda & , \quad w_i^* < -\lambda \end{cases} \tag{14}$$

Solution

Since $E[X + Y] = E[X] + E[Y]$ we can combine (3), (12) together with the expected value of w_i^* when $\sigma \to 0$ which is $\mathbf{x}_i^T \mathbf{t}$

Case 1: $w_i^* > \lambda$

$$E\Big[\hat{w}_i^{(1)} - w_i^*\Big] = \mathbf{x}_i^T \mathbf{t} - \frac{\lambda \mathbf{x}_i^T \mathbf{t}}{|\mathbf{x}_i^T \mathbf{t}|} - \mathbf{x}_i^T \mathbf{t} = \frac{\lambda \mathbf{x}_i^T \mathbf{t}}{|\mathbf{x}_i^T \mathbf{t}|} = -\lambda \tag{15}$$

Case 2: $w_i^* < -\lambda$

$$E\Big[\hat{w}_i^{(1)} - w_i^*\Big] = \mathbf{x}_i^T \mathbf{t} - \frac{\lambda \mathbf{x}_i^T \mathbf{t}}{|\mathbf{x}_i^T \mathbf{t}|} - \mathbf{x}_i^T \mathbf{t} = \frac{\lambda \mathbf{x}_i^T \mathbf{t}}{|\mathbf{x}_i^T \mathbf{t}|} = \lambda \tag{16}$$

Case 3: $|w_i^*| \leq \lambda$

$$E\Big[\hat{w}_i^{(1)} - w_i^*\Big] = -\mathbf{x}_i^T \mathbf{t} = -w_i^* \tag{17}$$

The acronym LASSO stands for "Least Absolute Shrinkage and Selection Operator". Unlike Ridge regression which only shrinks the values. LASSO uses a form of "selection" since it forces low values of w to go to zero.

Hyperparameter-learning via K-fold cross-validation

In this section, data points have been simulated from:

$$t(n) = f(n) + \sigma e(n) \tag{18}$$

$$f(n) = \mathbb{R}(5e^{i2\pi(\frac{n}{20}+\frac{1}{3})} + 2e^{i2\pi(\frac{n}{5}-\frac{1}{4})}) \tag{19}$$

$$e(n) \sim N(0,1) \tag{20}$$

The regression matrix consists of 500 sine and cosine pairs.

Task 4

Below are the plots for the reconstructed point together with the original data as well as the interpolation of the reconstructed points $\lambda = 0.1$, $\lambda = 10$ and $\lambda = 3$. As we can see in 1, with a small lambda, the model is overfitted since it basically moves through every data point we have. When using a large λ like 10, the model is too simple and we have large difference between the original data points and the reconstructed ones. This is seen in 2 This implies that there exists a λ somewhere between 0.1 and 10. Figure 3 shows $\lambda = 3$ which gives a relatively well fitted curve without adhering to every data point as with $\lambda = 0.1$.

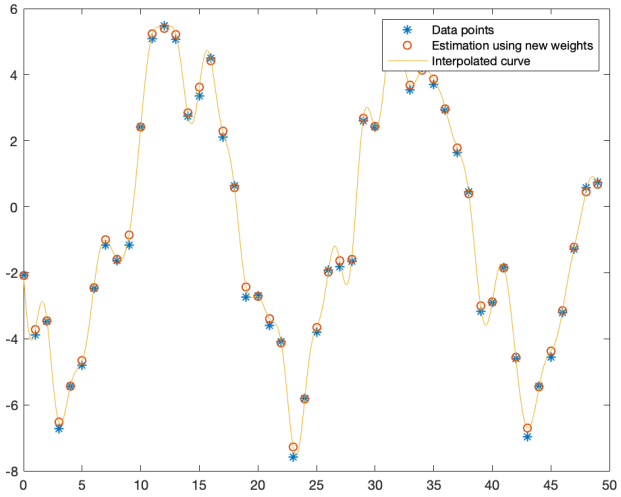

Figure 1: The reconstruction using $\lambda = 0.1$. The blue stars show the original data and the line shows the recreated estimations and the interpolated line.

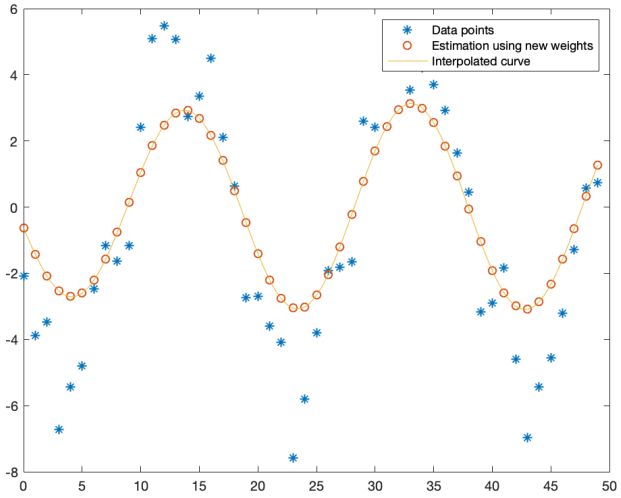

Figure 2: The reconstruction using $\lambda = 10$. The blue stars show the original data and the line shows the recreated estimations and the interpolated line.

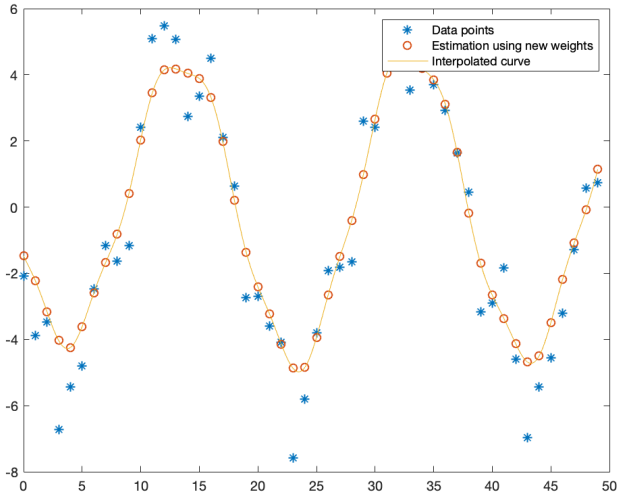

Figure 3: The reconstruction using $\lambda = 3$. The blue stars show the original data and the line shows the recreated estimations and the interpolated line.

4.1

Comparing the amount of parameters differing from zero between the different estimations we see that an increased λ means that more weights are zero. Below, W is the amount of nonzero weights.

$$\lambda = 0.1 \rightarrow W = 276 \tag{21}$$
$$\lambda = 10 \rightarrow W = 6 \tag{22}$$
$$\lambda = 0.1 \rightarrow W = 16 \tag{23}$$

The actual number of nonzero coordinates needed to model the data given the true frequencies is four. From the numbers above it seems that the best solution requires 16 parameters. This might be due to the relatively low number of data points. When $\lambda = 10$ we see that the number of nonzero weights are 6, which is still more than what was needed to model the data given the true frequencies. However looking at 2 we see that the result is an oversimplification. If given a denser amount of data points, the number of nonzero parameters should decrease as the actual frequencies would show more clearly.

Task 5

The task here is to instead of guessing between which λ will give the best prediction. We use k folds cross validation to loop over the data set and different lambdas chosen in advance. I used $K = 20$ and 20 different lambdas ranging from 0.01 to 15. Figure 4 below show the $\lambda(= 2.1892)$ which gives the least validation error. Since the lambda vector is chosen beforehand, an idea is to perform cross validation again with new lambdas around the best current value to find a slightly more optimal λ. In figure 5 we can see how the estimations and validation error changes with λ.

Looking at figure 4 the data does however not seem to be a perfect fit as an increase in the amplitude of the curve would reach the lower and higher points. I suspect there might be some error in the implementation that causes this.

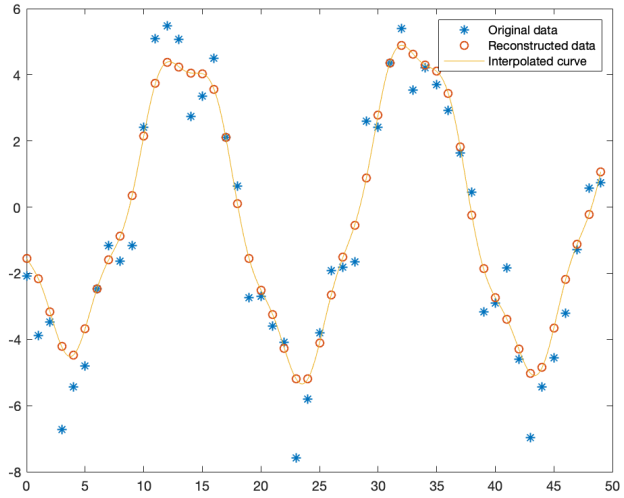

Figure 4: The reconstruction using $\lambda = 2.1892$. The blue stars show the original data and the line shows the recreated estimations and the interpolated line.

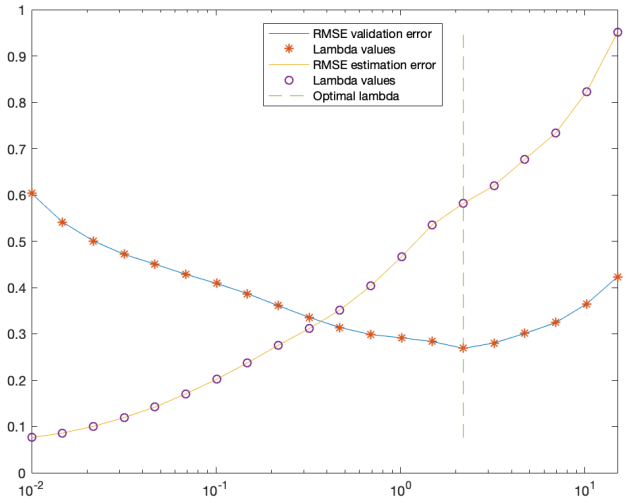

Figure 5: The x-axis is logarithmic to show an even distribution between the lambdas. The validation and estimation error after 20-folds cross validation using 20 different lambdas. The vertical line shows the optimal $\lambda = 2.1892$ which is where the validation error reaches its minimum.

Denoising of an audio excerpt

In this section we'll take a noise audio excerpt and use LASSO-regression to try to denoise the sample. The total sound clip is 5 seconds long and is divided into a training and a test set. The test set will not be involved in the cross validation but only in the testing phase.

Sounds of importance often have a narrow-band of frequencies and noise often contains many frequencies. The point with LASSO-regression in this case is to try and find the strong frequencies and zero out the others. Since the sample size is large, a division of the data will be made into 40ms and the sequences will be estimated separately.

Cross validation will, be made as in the previous task but we'll estimate a single λ for all different data sections. Since the time it takes for the calculation severely increases, the number of folds is decreased to 8 and the number of lambdas to 10.

Task 6

The data training data was divided into shorter lengths of 352. For each section, a cross validation is made and the estimation and validation error for every fold and λ is summed for all sections. Eventually the λ which gives the least prediction error is used to try and denoise the test data Ttest. The λ ranged form 0.0001 to 5 and the optimal was found to be at $\lambda = 0.0037$. This is seen in figure 6.

As seen in 5 the smallest validation error comes a bit after the estimation and validation error lines cross. The λ found is relatively small compared to the previous task which implies this problem required more nonzero weights to model the data.

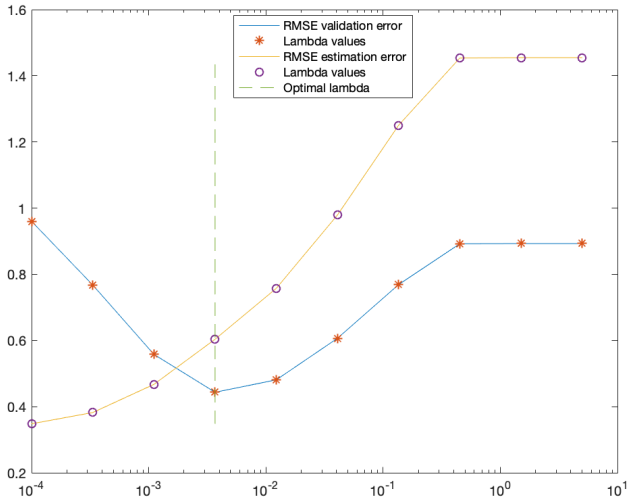

Figure 6:

Task 7

Using the optimal λ aquired in the previous task, the noise of the original noisy signal doesn't seem to change at all. I am not quite sure why this is the case.

However, during a previous run with larger lambdas my optimal was found to be at $\lambda = 0.01$. When plotting as in figure 6 I saw that it was not a minimum point so the range of λ was lowered. When I listened to the denoised signal using $\lambda = 0.01$ the results were much better than with the "optimal" λ I had acquired.

This might have something to do with my suspicion mentioned in task 5 where the reconstructed curve didn't reach the higher and lower points. I find it strange that a minimisation of validation error using cross validation would yield a λ which is worse in the reconstruction than a somewhat trivial one. This would maybe be the case if the testing data differs very much from the training data.

The sound for the optimal λ is in the attached file denoised_audio.mat and the better sounding audio for $\lambda = 0.01$ is in denoised_audio_lambda0.01.mat.