Machine Learning Assigment 1

Jesper Laurell

April 2020

1 Penalized regression via the LASSO

Task 1

$$minimize \frac{1}{2}||r_i - x_i w_i||_2^2 + \lambda |w| \tag{1}$$

The objective to minimize the above function can be achieved by differentiating with respect to w_i and setting equal to 0.

$$(-x_i^T)(r_i - x_i w_i) + \lambda \frac{w_i}{|w_i|} = 0 \tag{2}$$

$$w_i = \frac{x_i^T r_i - sign(w_i)\lambda}{x_i^T x_i} \tag{3}$$

From equation 3 we can use the fact that $\lambda > 0$ and $x_i^T x_i > 0$ to realize that $sign(w_i) = sign(x_i^T r_i) = \frac{x_i^T r_i}{|x_i^T r_i|}$. Putting this result into 3 yields the following:

$$w_i = \frac{x_i^T r_i - sign(x_i^T r_i)\lambda}{x_i^T x_i} \tag{4}$$

$$w_i = \frac{x_i r_i - \frac{x_i^T r_i}{|x_i^T r_i|}\lambda}{x_i^T x_i} \tag{5}$$

$$w_i = \frac{x_i r_i}{x_i^T x_i |x_i^T r_i|}(|x_i^T r_i| - \lambda) \tag{6}$$

This is equal to the given equation and the proof is complete.

Task 2

Starting from 6 with the given condition $|x_i^T r_i^{(j-1)}| > \lambda$ and adding the equation $r_i = t - \sum_{l \neq i} x_l w_l$ we achieve the following.

$$w_i^{(j)} = \frac{x_i^T (t - \sum_{l \neq i} x_l w_l)}{x_i^T x_i |x_i^T (t - \sum_{l \neq i} x_l w_l)|}(|x_i^T (t - \sum_{l \neq i} x_l w_l| - \lambda)) \tag{7}$$

The given fact that the regression matrix X is orthogonal means that the dot product of two different vectors from it will always equal to 0. With the help of this the expression can be greatly simplified. All sums can be zeroed out since each term from the sums will contain a factor of a vector orthogonal to x_i^T and each sum will be multiplied with x_i^T. Also using the fact that a vector times its transpose is the The resulting expression will be as follows:

$$w_i^{(j)} = \frac{x_i^T t}{|x_i t|}(|x_i^T t| - \lambda) \tag{8}$$

This expression is independent of j which means the iteration does not matter and thus: $w_i^{(1)} = w_i^{(2)}$, and the proof is complete.

Task 3

Continuing from the results in Task 2 above equation 8 is still valid for this task since X is orthogonal. Also, we have to consider both parts of the original given equation for w_i.

$$w_i^{(j)} = \begin{cases} \frac{x_i^T t}{x_i^T x_i |x_i^T t|}(|x_i^T t| - \lambda), & |x_i^T r_i^{(j-1)}| > \lambda \\ 0, & |x_i^T r_i^{(j-1)}| \leq \lambda \end{cases} \tag{9}$$

From the first part of the above we can use the other given equation:

$$\mathbf{t} = \mathbf{X}\mathbf{w}^* + \mathbf{e}, \mathbf{e} \sim \mathcal{N}(\mathbf{0}_N, \sigma \mathbf{I}_N). \tag{10}$$

This means for each iteration that:

$$t = x_i w_i^* + e \tag{11}$$

LHS of the given equation that needs to be proven is the following:

$$\lim_{\sigma \to 0} E(w_i^{(1)} - w_i^*) \tag{12}$$

However letting $\sigma \to 0$ means that $e \to 0$ and with the use of equation 11 the original expression for $w_i^{(j)}$ in 9 can be simplified as follows:

$$w_i^{(j)} = \frac{x_i^T (x_i w_i^* + e)}{|x_i^T (x_i w_i^* + e)|}(|x_i^T (x_i w_i^* + e)| - \lambda), \qquad |x_i^T r_i| > \lambda \tag{13}$$

$$w_i^{(j)} = \frac{x_i^T (x_i w_i^*)}{|x_i^T (x_i w_i^*)|}(|x_i^T (x_i w_i^*)| - \lambda), \qquad |x_i^T r_i| > \lambda \tag{14}$$

$x_i^T x_i = I$ yields:

$$w_i^{(j)} = \frac{w_i^*}{|w_i^*|}(|w_i^*| - \lambda), \qquad |x_i^T r_i| > \lambda \tag{15}$$

$$w_i^{(j)} - w_i^* = -\lambda * sign(w_i^*), \qquad |x_i^T r_i| > \lambda \tag{16}$$

$$\lim_{\sigma \to 0} E(w_i^{(1)} - w_i^*) = E(-\lambda * sign(w_i^*)) = \begin{cases} -\lambda, & w_i^* > \lambda \\ \lambda, & w_i^* \leq -\lambda \end{cases} \quad (17)$$

Note that this requires the given fact that $0 \leq \lambda$.

The final case when $|x_i r_i| < \lambda$ can be considered separately since for this case the equation $w_i^{(j)} = 0$ hold true. This means that $\lim_{\sigma \to 0} E(w_i^{(1)} - w_i^*) = E(0 - w_i^*) = -w_i^*$. Thus also the final part of the equation is proven.

2 Hyperparameter-learning via K-fold cross-validation

Task 4

For this task the provided skeleton-file was filled out to run the coordinate descent algorithm and three different lambdas were tested accompanied with one plot each. It turns out that the two suggested lambdas 0.1 and 10 resulted in one somewhat overfitted model and one underfitted model respectively. The custom value for lambda was set to 5 resulting in the best option out of the three. The

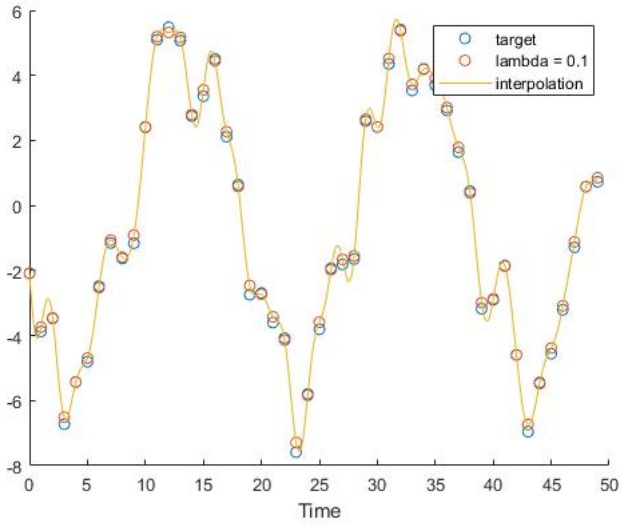

Figure 1: Overfitting

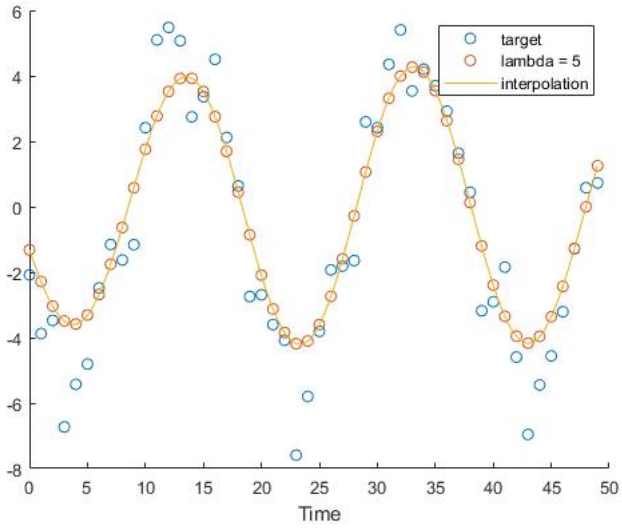

Figure 2: Lambda set to 5 for compromise between underfitting and overfitting.

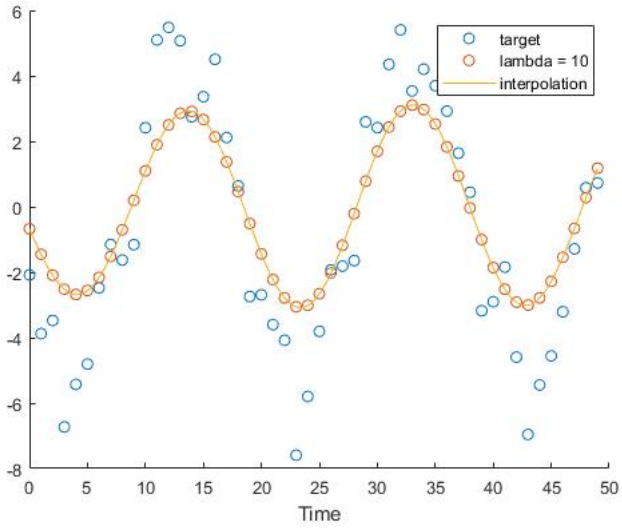

Figure 3: Underfitting

For the second subtask the resulting weights were analyzed by checking how many non-zero elements each vector contained. The result is presented in the table below.

λ	non-zero elements
0.1	223
5	10
10	6

Table 1: Number of non-zero elements for the three different lambdas

It is clear that with a larger lambda the number of zeroes increase. This is because the larger the lambda the more impact it has to cancel out parameters that do not seem to be important and set them to zero. This is both a strength and a weakness of the LASSO approach. For sets with a lot of parameters that are irrelevant it is very useful to be able to completely nullify them, but for sets where few parameters are irrelevant information will be overlooked when setting the impact if relevant parameters to zero.

Task 5

The skeleton file was implemented and 50 lambdas were considered with 10 folds for cross validation. The estimation and validation were analyzed in a random order to avoid biasing.

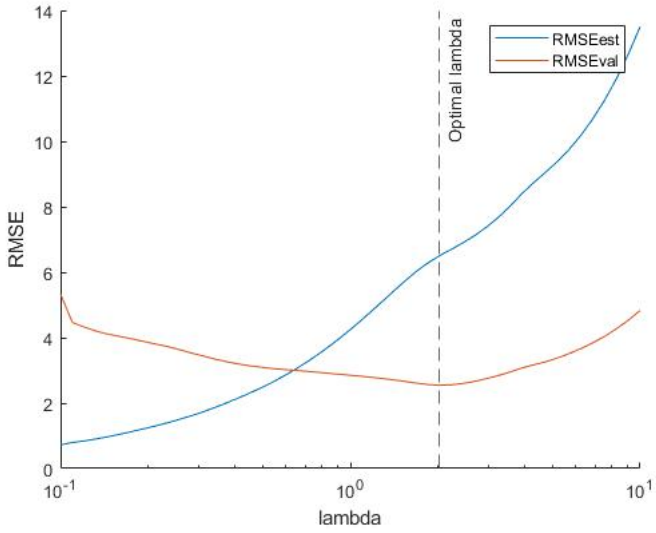

Figure 4: The validation error and estimation error for different lambdas. Optimal lambda found at the vertical line, minimizing the validation error.

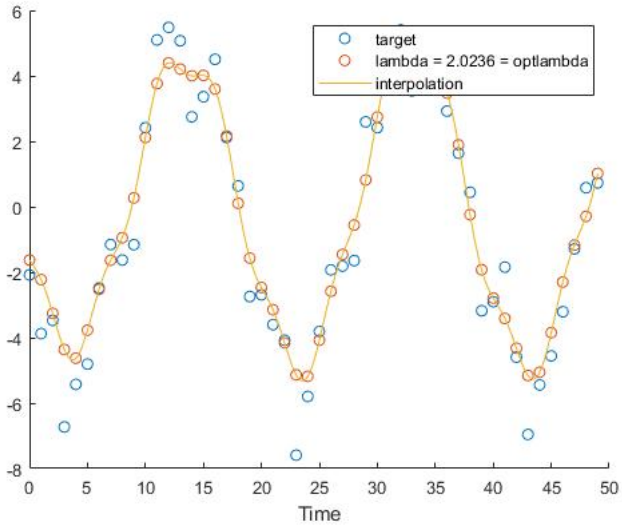

Figure 5: Interpolation when using the optimal lambda 0.0236

As seen in figure 4 the estimation error is growing with an increasing lambda. This is because small lambdas mean the generated model will be very biased toward the training data and the training error (= estimation error) will be small. However for larger lambdas the model will be less biased to the training data and the estimation error will grow. However the interesting error is really the validation error, which has a sweet spot when choosing lambda. The validation error is more relevant because this is how the model responds to unseen data which is what a useful model will be used for. This error is large for small lambdas (underfit), reaches a minimum (optimal fit), and then grows again (overfit).

A human would probably consider this interpolation somewhat overfit since it does not very accurately represent a regular sinus wave. However a human is often not the best judge when it comes to choosing optimal parameters and with our criterion for choosing an optimal lambda this is the result when plotting with a lambda giving the least validation error.

3 Denoising of an audio excerpt

Task 6

The approach for solving task 6 was much like solving task 5. The multiframe script was filled in with similar logic used in task 5, only this time looping also over the frames which added an outer loop.

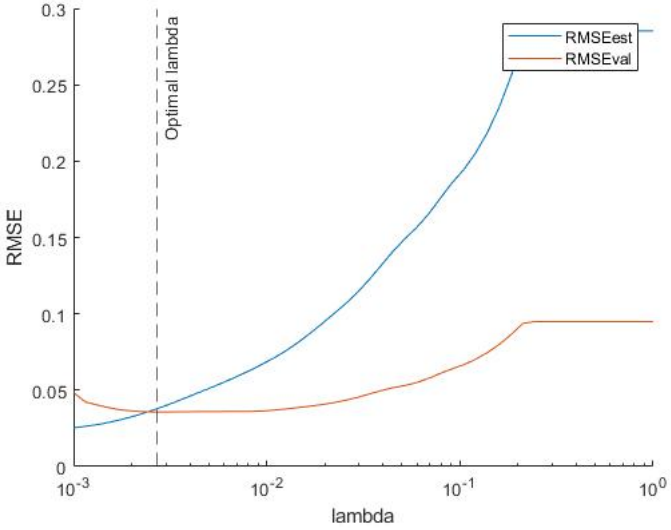

Figure 6: Finding the optimal lambda for the audio file.

The resulting optimal lambda was 0.0027 and similarly to task 5 it was found where the validation error reached a minimum.

Task 7

Denoising the audio with our found optimal lambda it is clear that some of the noise was removed, but there is still a lot of background noise remaining. For a human ear it does not sound great and personally i would consider it "better cleaned" with a bit larger lambda which would underfit the data but clean out more of the noise. For example setting lambda to 0.01 was in my opinion a better sounding output.