Assignment 1
FMAN45
Machine Learning

Penalized regression via the LASSO, Hyperparameter-learning via K-fold cross-validation, and Denoising of an audio excerpt

Author

Kajsa Hansson Willis

Lund University

Lunds universitet
Lunds Tekniska Högskola

Spring 2025

1 Exercise 1

In the first exercise, the purpose is to verify Equation 1 by solving Equation 2 for $w_i \neq 0$. In the equation and the assignment, w_i represents the explanatory variable, or the weights, given a linear relationship to the response variable, or the data, t. X is the regression matrix, which is the hypothesis space, and λ is the capacity hyperparameter that is $\lambda \geq 0$.

$$\hat{w}_i^{(j)} = \frac{x^T r_i^{(j-1)}}{x_i^T x_i |x_i^T r_i^{(j-1)}|}(|x_i^T r_i^{(j-1)}| - \lambda) \text{ where } r_i^{(j-1)} = t - \sum_{l<i} x_l \hat{w}_l^{(j)} - \sum_{l>i} x_l \hat{w}_l^{(j-1)} \tag{1}$$

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

In order to solve the minimization expression, Equation 1, the derivative is taken and set to zero. This is done in Equation 3, which uses the fact that $\frac{d|w_i|}{dw_i} = \frac{w_i}{|w_i|}$ (since it is not zero) and the definition of the Euclidean norm.

$$\begin{aligned} \frac{d}{dw_i}(\frac{1}{2}||r_i - x_i w_i||_2^2 + \lambda |w_i|) &= \lambda \frac{w_i}{|w_i|} + \frac{d}{dw_i}(\frac{1}{2}(r_i - x_i w_i)^T (r_i - x_i w_i)) \\ &= \lambda \frac{w_i}{|w_i|} + \frac{d}{dw_i}(\frac{1}{2}(r_i^T r_i - r_i^T x_i w_i - x_i^T w_i r_i + x_i^T w_i x_i w_i)) \\ &= \lambda \frac{w_i}{|w_i|} - \frac{1}{2} r_i^T x_i - \frac{1}{2} x_i^T r_i + x_i^T x_i w_i = 0 \end{aligned} \tag{3}$$

Using Equation 3 to isolate the optimal w_i, w_i^* through matrix-multiplication gives the first expression in Equation 4. As $r_i^T x_i$ and $x^T r_i$ are scalar values and equal each other, the expression can be rewritten as is done in Equation 4. The expressions for w_i^* when it is positive and negative are Equation 5 and 6, respectively.

$$w_i^* = \frac{0.5 r_i^T x_i + 0.5 x^T r_i - \lambda \frac{w_i}{|w_i|}}{x_i^T x_i} = \frac{x^T r_i - \lambda \frac{w_i}{|w_i|}}{x_i^T x_i} = \frac{x^T r_i - \lambda * \text{sgn}(w_i)}{x_i^T x_i} \tag{4}$$

$$w_i^* = \frac{x^T r_i - \lambda}{x_i^T x_i}, w_i > 0 \tag{5}$$

$$w_i^* = \frac{x^T r_i + \lambda}{x_i^T x_i}, w_i < 0 \tag{6}$$

Another way to express the optimal w_i^* is through Equation 7. As the expression in the parenthesis must be positive, considering it is the sum of positive parameters ($\lambda \geq 0$, absolute values, and a square), the following must hold sgn $x_i^T r_i =$ sgn w_i. Thus, Equation 8 is derived.

$$x_i^T r_i = w_i(\lambda \frac{1}{|w_i|} + x_i^T x_i) \tag{7}$$

$$w_i^* = \frac{x^T r_i - \lambda * \text{sgn}(x_i^T r_i)}{x_i^T x_i} = \frac{x^T r_i - \lambda * \frac{x_i^T r_i}{|x_i^T r_i|}}{x_i^T x_i} = \frac{x^T r_i * \frac{|x_i^T r_i|}{|x_i^T r_i|} - \lambda * \frac{x_i^T r_i}{|x_i^T r_i|}}{x_i^T x_i} = \frac{x^T r_i(|x_i^T r_i| - \lambda)}{x_i^T x_i |x_i^T r_i|} \tag{8}$$

This verifies that the two expressions are equal if $w_i \neq 0$.

2 Exercise 2

In this task, the aim was to show that the coordinate descent solver converges in at most one full pass over the coordinates in w given the orthogonal regression matrix. This is done by showing $\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = 0, \forall i$. The orthogonal regression matrix is such that $X^T X = I_N$, where I_N is an NxN identity matrix. This implies that $x^T x = 1$ in Equation 9.

There are two possible cases for this. In the first case $|x_i^T r_i^{(j-1)}| \leq \lambda$ and $\hat{w}_i^{(j)} = 0$. It is then straightforward to show that $w_i^{(2)} - \hat{w}_i^{(1)} = 0 - 0 = 0$. Thus, convergence has been shown for the first case.

The second case, where $|x_i^T r_i^{(j-1)}| > \lambda$ is more complicated. In this case, the value of $\hat{w}_i^{(j)}$ is determined by Equation 9 and 10. The full derivation is shown in Equation 11. An important point is that the orthogonal condition implies that the scalar products of x_i and x_l are zero for all cases where $i \neq l$ (which is all the vectors in Equation 10.

$$\hat{w}_i^{(j)} = \frac{x_i^T r_i^{(j-1)}}{x_i^T x_i |x_i^T r_i^{(j-1)}|} * (|x_i^T r_i^{(j-1)}| - \lambda) \tag{9}$$

$$r_i^{(j-1)} = t - \sum_{l<i} x_l \hat{w}_l^{(j)} - \sum_{l>i} x_l \hat{w}_l^{(j-1)} \tag{10}$$

$$\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})}{x_i^T x_i |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})|} * (|x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})| - \lambda)$$
$$- \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})}{x_i^T x_i |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})|} * (|x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})| - \lambda) =$$
$$= \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})}{x_i^T x_i} - \lambda * \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})}{x_i^T x_i |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})|}$$
$$-(\frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})}{x_i^T x_i} - \lambda * \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})}{x_i^T x_i |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})|})$$
$$= \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})}{1} - \lambda * \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})}{I_N |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(2)} - \sum_{l>i} x_l\hat{w}_l^{(1)})|}$$
$$-(\frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})}{1} - \lambda * \frac{x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})}{1 |x_i^T(t - \sum_{l<i} x_l\hat{w}_l^{(1)} - \sum_{l>i} x_l\hat{w}_l^{(0)})|})$$
$$= x_i^T * t - \lambda * \frac{x_i^T * t}{|x_i^T * t|} - (x_i^T * t - \lambda * \frac{x_i^T * t}{|x_i^T * t|}) = 0 \tag{11}$$

Thus, it has been found that the two terms equal each other, so the expression is equal to zero. Therefore, the convergence of the coordinate descent solver has been shown.

3 Exercise 3

In this exercise, the LASSO estimate's bias for an orthogonal regression matrix and data generated by Equation 12 is explored when $\sigma \to 0$. The bias is expressed by Equation 13.

$$t = Xw^* + e,\ e \sim N(0_N, \sigma I_N) \tag{12}$$

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) \tag{13}$$

As the regression matrix is orthogonal, it is known that the weights can be described by Equation 14 if $|x_i^T r_i^{(j-1)}| > \lambda$ as the sum terms in r_i multiply to zero when inserted.

$$\hat{w}_i^{(j)} = x_i^T * t - \lambda * \frac{x_i^T * t}{|x_i^T * t|} \tag{14}$$

This can be further divided into two cases, one where $x_i^T r_i^{(j-1)} > \lambda$ and one where $x_i^T r_i^{(j-1)} < -\lambda$. These cases lead to Equation 15 and 16, respectively.

$$\hat{w}_i^{(j)} = x_i^T * t - \lambda \tag{15}$$

$$\hat{w}_i^{(j)} = x_i^T * t + \lambda \tag{16}$$

It is known that the relationship in Equation 12 holds true. This can be rewritten according to Equation 17. Thus, w_i^* is described by Equation 18.

$$X^{-1}t - e = w^*,\ e \sim N(0_N, \sigma I_N) \tag{17}$$

$$x_i^{-1}t - e_i = w_i^*,\ e_i \sim N(0_{iN}, \sigma I_{iN}) \tag{18}$$

Dividing up the solutions into the three different cases:

Case 1: $x_1^T r_i^{(j-1)} > \lambda$

In the first case, the expression in Equation 19 leverages the fact that in an orthogonal matrix, the inverse is equal to its transpose. The limit of the expression is then explored in Equation 20.

$$E(\hat{w}_i^{(1)} - w_i^*) = E((x_i^T * t - \lambda) - (x_i^{-1} * t - e_i)) = E(x_i^T * t - \lambda - x_i^T * t + e_i) = E(-\lambda + e_i) \tag{19}$$

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = \lim_{\sigma \to 0} E(-\lambda + e_i) = -\lambda \tag{20}$$

Case 2: $|x_1^T r_i^{(j-1)}| \leq \lambda$

In the second case, the value of $w_i^{(1)}$ is 0. The limit can be expressed with Equation 21.

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = \lim_{\sigma \to 0} E(0 - w_i^*) = -w_i^* \tag{21}$$

Case 3: $x_1^T r_i^{(j-1)} < -\lambda$

The third case is similar to the first case, and the same concepts are applied, but this case follows Equation 22 instead. The limit is then determined by Equation 23.

$$E(\hat{w}_i^{(1)} - w_i^*) = E((x_i^T * t + \lambda) - (x_i^{-1} * t - e_i)) = E(x_i^T * t + \lambda - x_i^T * t + e_i) = E(\lambda + e_i) \tag{22}$$

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = \lim_{\sigma \to 0} E(\lambda + e_i) = \lambda \tag{23}$$

As the expression $x_1^T r_i^{(j-1)}$, that comes from the limits given in the assignment, can be rewritten as $x_1^{-1}t$ from the orthogonality property as the transpose is equal to the inverse and the product of all the sums from the expression for $r_i^{(j-1)}$ and x_i^T become zero and thus t is the only non-zero term left. It is previously known from Equation 18 that this expression is equal to w_i^*. This means

that the limits can be rewritten as $w_i^* > \lambda$, $|w_i^*| \leq \lambda$, and $w_i^* < -\lambda$, respectively. This means that the following holds true:

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

Lasso is the least absolute shrinkage and selection operator and implements a penalty for variables in order to avoid unnecessary coordinates. When $|w_i^*|$ is smaller than λ, it is set to zero, which is where the selection operator can add efficiency. The least absolute part signifies that it is a minimization problem with absolute values, or norms, which is apparent when observing Equation 2. The shrinkage comes from the λ-terms that are subtracted or added as bias, which can be observed in the expression above. Thus, the bias shrinks the LASSO estimate closer to zero by either $-\lambda$ if it is positive or λ if it is negative.

4 Exercise 4

In the fourth exercise, the aim is to implement a cyclic coordinate descent solver for the coordinate-wise LASSO, producing reconstruction plots for a selection of λ-values, and counting the number of non-zero coordinates for the different λ-values. An example of a good reconstruction plot was given in the assignment and can be viewed in Figure 1.

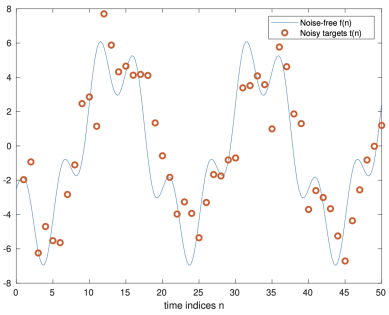

Figure 1: Given figure from the assignment for comparison: Plot of the provided noisy target values (data) t, together with the noise-free data generating function f(n) in (10). A good reconstruction should look somewhat close to this!

First, the LASSO solution was constructed by completing the given *lasso_ccd* Matlab function. In order to make it function, it was ensured that the current regression vector was selected, the impact of the old weight vector was put into the residual, the lasso estimate was updated according to Equation 1 or zero when applicable, and the impact of the newly estimated weight was removed from the residual again.

The selected λ-values are 0.1, 10, and 0.75. An array was created with these values that was then inserted into the *lasso_ccd* function. For each of these values of λ and the 50 data points, the reconstructed data points, y, were calculated through $y = X * \hat{w}$. Additionally, an interpolated reconstruction of the data was calculated using a highly sampled regression matrix. These were then plotted together with the original data points, t.

The plot for $\lambda = 0,1$ can be seen in Figure 2. It is quite similar to the noise-free f(n) in Figure 1, but a little more sharp, which might indicate it is overfitted to the points. The real data is very close to the reconstructed data points.

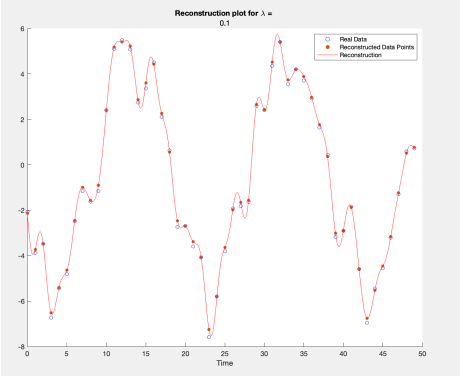

Figure 2: Reconstruction plot for $\lambda = 0,1$ with the real data, the reconstructed data points, and the reconstruction for the 50 data points

The plot for $\lambda = 10$ can be seen in Figure 3 and resembles a sinusoid. It is dissimilar to the noise-free f(n) in Figure 1 as the shapes differ to a large extent. It shows poor alignment with the real data as well, as there are considerable differences between the reconstructed data and the real data at the peaks. This indicates an underfit.

For the last plot, several values were tested and it was decided that 0,75 was the most optimal. The reconstruction plot can be viewed in Figure 4. This plot is more of a middle-ground when it comes to underfitting or overfitting the plot, and it is quite similar to Figure 1.

Only four non-zero coordinates are required. This can be compared to the amount of non-zero coordinates for the different λ-values, which can be viewed in Table 1. Clearly, a value of 10 is the closest to the true value, with only six non-zero coordinates. However, it was established that this was clearly an underfit, which indicates that even if the underlying signal is not very complex, the noise increases the complexity needed for fitting considerably. The lowest λ-value, 0,1, has the highest number of non-zero parameters and was overfitted, as expected given the regularization behavior. The more balanced λ-value, 0,75, has 79 parameters, which is a considerable jump from 4, but still significantly lower than 224.

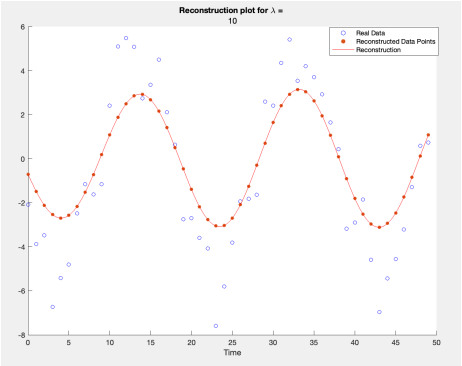

Figure 3: Reconstruction plot for $\lambda = 10$ with the real data, the reconstructed data points, and the reconstruction for the 50 data points

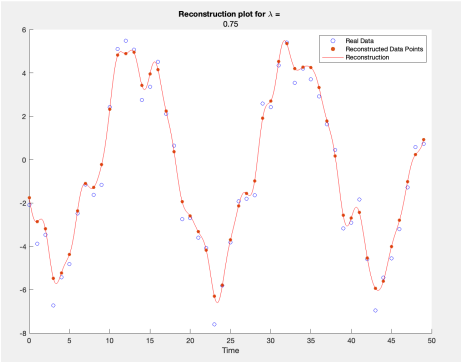

Figure 4: Reconstruction plot for $\lambda = 0,75$ with the real data, the reconstructed data points, and the reconstruction for the 50 data points

Lambda:	**Non-zero coordinates:**
0,1	224
10	6
0,75	79
True value:	**4**

Table 1: The amount of non-zero coordinates for different hyperparameters (λ-values) as well as the amount of non-zero coordinates needed to model the true data

5 Exercise 5

In the fifth exercise, the purpose was to implement a K-fold cross-validation scheme for the LASSO solver implemented in the fourth exercise.

This was done by finishing the given function *lasso_cv* which calculates the

LASSO solution problem and trains the hyperparameter using cross-validation. This function leveraged the previously completed *lasso_ccd* function. The root mean squared error (RMSE) was calculated according to Equation 24 for both the validation data and the estimation data for a certain value of λ where K represents the number of folds, N is the size of the data set, and I is the indices for the data set, and w is the weight estimate.

$$RMSE(\lambda_j) = \sqrt{K^{-1} \sum_{k=1}^{K} N^{-1} ||t(I) - X(I)\hat{w}^{[k]}(\lambda_j)||_2^2} \tag{24}$$

With this function, the RMSE-values for an array of 20 equally spaced values of λ starting with 0.001 up to $\lambda_{max} = \max |X' * t| = 24.5742$ were calculated for both the validation data and the estimation data. These were then plotted in Figure 5 together with the optimal value of λ_{opt}. It is clear that the optimal value of λ is the lowest point of the RMSE for the validation data. This value corresponds to a λ-value of 1.812.

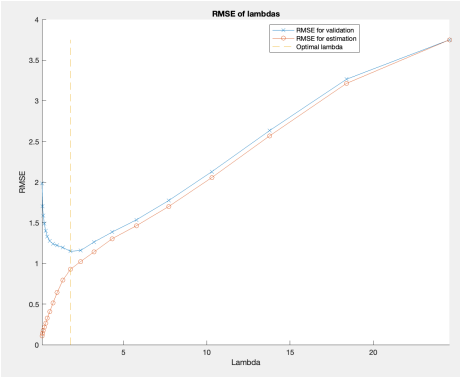

Figure 5: Plot of the root mean square error (RMSE) on validation and estimation data of different values of λ, and the optimal $\hat{\lambda}$

Further, a reconstruction plot was created. This was done in the same way as in Exercise 4, with $y = X * \hat{w}_{opt}$ for both the given regression matrix and the highly sampled regression matrix. These were then plotted with the true values, and can be viewed in Figure 6.

Comparing the reconstruction plot to Figure 1, it is clear that they are quite similar. This indicates, that the reconstruction is quite good and that the K-fold cross-validation scheme for the LASSO solver improves the estimate as the value of λ now can be chosen more favorably.

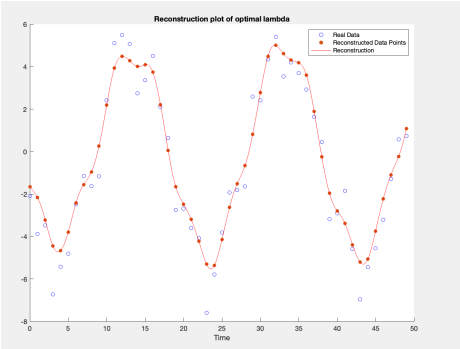

Figure 6: The reconstruction plot in exercise 5 with the real data, the reconstructed data points, and the reconstruction for the 50 data points for $\hat{\lambda} = 1,812$

6 Exercise 6

In the sixth exercise, the aim was to implement a K-fold cross-validation scheme for a multi-frame audio excerpt of piano music modified to find an optimal $\hat{\lambda}$ for all frames.

An incomplete matlab function, *multiframe_lasso_cv*, was provided for the exercise. The first step was to complete it, which was done similarly to in exercise 5 by determining appropriate indices based on the lasso-framework, determining an $\hat{w}$, calculating the mean errors over the frames, and finding the optimal $\hat{\lambda}$ and $\hat{w}$.

The array of λ-values was selected as an equally spaced array of 20 values starting from 0,0001 to the λ_{max} which was calculated by dividing the T_{train} vector into 56 frames with 352 values and calculating the λ_{max} of each one according to $\max X'_{audio} * T_{train,frame}$ and then taking the largest one, which was 0,7517. The number of folds, the K-value, was selected as 5 for this exercise.

These values were then inserted into the *multiframe_lasso_cv*. This resulted in estimations of the optimal weights, the optimal $\hat{\lambda}$, the RMSE of the validation data, and the RMSE of the estimation data.

Then, a plot was created, see Figure 7, of the RMSE values for the different values of λ and the optimal value of $\hat{\lambda}$, which was calculated to 0,0049. In order to observe the critical area better, another zoomed in plot was also constructed, which is seen in Figure 8. Clearly, the optimal $\hat{\lambda}$ is the lowest point on the validation RMSE plot. The plot also shows that for the validation data, the error is quite high for very low λ-values and decreases when λ increases until it reaches the optimum. Then, the error increases significantly, but eventually flattens out. The error for the estimation data increases with the hyperparameter, but the increase is diminishing. This is very much expected since the hyperparameter represents how significant a factor must be to not be ignored, so a higher value leads to more ignored factors for the estimation data. It is not unexpected for the validation data either, as an appropriate $\hat{\lambda}$-value is important in order to

avoid either overfitting or underfitting the data. However, when dividing up the data into frames, the optimal $\hat{\lambda}$-value differs considerably compared to in exercise 5, where the optimal value is found to be 1,812, although the difference is not unexpected, as the multi-frame method handles smaller amounts of data as it is divided up.

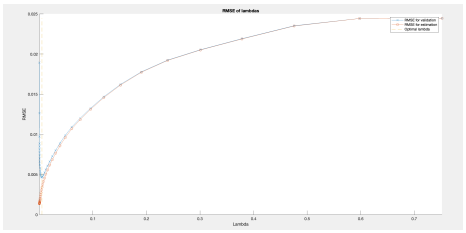

Figure 7: Plot of the root mean square errors of the validation data and the estimation data with the optimal $\hat{\lambda}$ from exercise 6

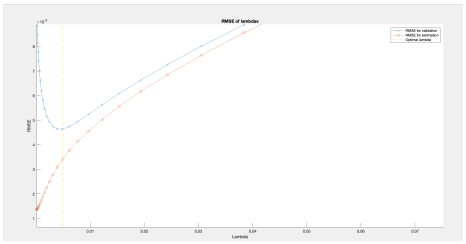

Figure 8: Zoomed in plot of the root mean square errors of the validation data and the estimation data with the optimal $\hat{\lambda}$ from exercise 6

7 Exercise 7

In the seventh exercise, the $\hat{\lambda}_{opt}$, 0,0049, from the sixth exercise was used to denoise the test data, which is a noisy recording of piano music. This was done using the provided Matlab function *lasso_denoise*. The file *denoised audio* with the resulting signals is provided in the report submission.

It must be prefaced that the author is tone deaf and therefore finds it particularly difficult to distinguish tones from noise in music. However, the results showed a clear improvement and efficient denoising with less background noise when using the optimal value of $\hat{\lambda}$. This implies that the λ-value seems to be quite appropriate.

Additionally, other values of λ were tested, including $\lambda = 0,04$ and $\lambda = 0,0004$ to understand what happens with a larger λ and a smaller λ. For the λ-values that were larger than the optimal one, there is even less background noise but it comes at the expense of the piano sound. For the smaller λ-values,

less background noise was removed. This further indicates that the optimal value of λ found is good.