Assigment 1 FMAN45

Kalle Josefsson ka5532jo-s

April 22, 2024

1 Task 1

The first task is to minimize the expression below with respect to 'w_i'

$$\min_w \left(\frac{1}{2} \|t - Xw\|_2^2 + \lambda \|w\|_1 \right) \tag{1}$$

The expression for the derivative is seen below.

$$\frac{\partial}{\partial w_i} \left(\frac{1}{2} \|t - Xw\|_2^2 + \lambda \|w\|_1 \right) \tag{2}$$

Now I do a simplification which simply is describing the derivative of the norm of w as in equation 3.

$$\frac{d|w_i|}{dw_i} = \frac{w_i}{|w_i|} \tag{3}$$

Now I want to differentiate the left expression, i.e the L2 norm with respect 'w_i'. The first thing I do is multiply the parenthesis and make a simplification since $r_i^T x_i = x_i^T r_i$ we get the expression.

$$\frac{1}{2} \left(r_i^T r_i - 2 w_i x_i^T r_i + w_i^2 x_i^T x_i \right) \tag{4}$$

Which we differentiate and since r_i and x_i have no w_i dependency the expression becomes very simple and the derivative of equation 2 becomes.

$$x_i^T (w_i x_i - r_i) + \lambda \frac{w_i}{|w_i|} \tag{5}$$

In order to minimize the expression we set the derivative to zero, but since we have an absolute value this becomes a little bit trickier and we introduce the two cases, $w_i < 0$ and $w_i > 0$. After moving w_i to the left hand side of the equation we get these two expression for the two cases.

$$\hat{w}_i^{(j)} = \begin{cases} \frac{x_i^T r_i^{(j-1)} - \lambda}{x_i^T x_i} & \text{if } w_i > 0 \\ \frac{x_i^T r_i^{(j-1)} + \lambda}{x_i^T x_i} & \text{if } w_i < 0 \end{cases} \tag{6}$$

In equation 6 we know that $x_i^T x_i$ is strictly positive hence we can change the conditions in equation 6 based on the numerator.

$$\hat{w}_i^{(j)} = \begin{cases} \frac{x_i^T r_i^{(j-1)} - \lambda}{x_i^T x_i} & \text{if } x_i^T r_i^{(j-1)} > \lambda \\ \frac{x_i^T r_i^{(j-1)} + \lambda}{x_i^T x_i} & \text{if } -x_i^T r_i^{(j-1)} < \lambda \end{cases} \tag{7}$$

We also know that lambda is strictly positive hence we can rewrite the two cases we can change the conditions again.

$$\hat{w}_i^{(j)} = \begin{cases} \frac{x_i^T r_i^{(j-1)} - \lambda \cdot \text{sgn}\left(x_i^T r_i^{(j-1)}\right)}{x_i^T x_i} & \text{if } \left|x_i^T r_i^{(j-1)}\right| > \lambda \\ 0 & \text{if } \left|x_i^T r_i^{(j-1)}\right| \leq \lambda \end{cases} \tag{8}$$

This can be rewritten on the desired form which is seen below.

$$\hat{w}_i^{(j)} = \begin{cases} \frac{x_i^T r_i^{(j-1)}}{x_i^T x_i \cdot \left|x_i^T r_i^{(j-1)}\right|} \left(\left|x_i^T r_i^{(j-1)}\right| - \lambda\right) & \text{if } \left|x_i^T r_i^{(j-1)}\right| > \lambda, \\ 0, & \text{if } \left|x_i^T r_i^{(j-1)}\right| \leq \lambda. \end{cases} \tag{9}$$

2 Task 2

In task two we were supposed to show that the coordinate descent solver will converge in at most one full iteration over w given that the regression matrix is orthogonal. In other words we are supposed to show that $\hat{w}_i^{(1)} = \hat{w}_i^{(2)}$. Given an orthogonal regression matrix we have $x_i^T x_j = 0$ if $j \neq i$. We also will use the real expression for $r_i^{(j-1)}$ which is presented below.

$$r_i^{(j-1)} = t - \sum_{\ell<i} x_\ell \hat{w}_\ell^{(j)} - \sum_{\ell>i} x_\ell \hat{w}_\ell^{(j-1)} \tag{10}$$

In the summationsigns we never have $l = i$ which means that when we multiply $r_i^{(j-1)}$ with x_i^T we only get $x_i^T t$ in the numerator and get the weight following weight updates.

$$\hat{w}_i^{(j)} = \begin{cases} \frac{x_i^T t}{x_i^T x_i \cdot \left|x_i^T t\right|} \left(\left|x_i^T t\right| - \lambda\right) & \text{if } \left|x_i^T t\right| > \lambda, \\ 0, & \text{if } \left|x_i^T t\right| \leq \lambda. \end{cases} \tag{11}$$

As we can see this has no j dependency but only depends on x_i^T, t and λ. Hence after every iteration j we will not get a different w_i since we have the same input arguments x_i^T, t and λ. Hence we have proven that $w_i^{(j)} = w_i^{(1)} = w_i^{(2)}$. Note that this only holds if the regression matrix in fact is orthogonal.

3 Task 3

Since we still have an orthogonal regression matrix we get the weight update according to the below expression.

$$\hat{w}_i^{(j)} = x_i^T t - \lambda \,\text{sgn}(x_i^T t) \tag{12}$$

Now we are going to see what happens when $\sigma \to 0$ and especially how it affects $x_i^T t$. So we have the expression.

$$\lim_{\sigma \to 0} x_i^T t = x_i^T X w^* + 0 = w^* \tag{13}$$

Now we have three different intervals look at, which are.

$$\text{Cases: } (1) x_i^T r_i^{(j-1)} > \lambda, \quad (2) x_i^T r_i^{(j-1)} < -\lambda, \quad (3) \left|x_i^T r_i^{(j-1)}\right| \leq \lambda. \tag{14}$$

For the three different cases presented we get three different expression for $\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*)$. For case (1) in equation (14) we get the following.

$$\begin{aligned}\lim_{\sigma\to 0} E(\hat{\omega}_i^{(1)} - \omega_i^*) \\ &= \lim_{\sigma\to 0} E\left(x_i^T t - \lambda \operatorname{sgn}(x_i^T t) - \omega_i^*\right) \\ &= E\left(\hat{\omega}_i^* - \operatorname{sgn}(\hat{\omega}_i^*) - \omega_i^*\right) \\ &= -\lambda E\left(\operatorname{sgn}(\omega_i^*)\right) \\ &= -\lambda\end{aligned}$$

For case (2) we use the same principle and get:

$$\begin{aligned}\lim_{\sigma\to 0} E(\hat{\omega}_i^{(1)} - \omega_i^*) \\ &= \lim_{\sigma\to 0} E\left(x_i^T t - \lambda \operatorname{sgn}(x_i^T t) - \omega_i^*\right) \\ &= E\left(\hat{\omega}_i^* - \operatorname{sgn}(\hat{\omega}_i^*) - \omega_i^*\right) \\ &= -\lambda E\left(\operatorname{sgn}(\omega_i^*)\right) \\ &= \lambda\end{aligned}$$

For the third case we get the simple equations:

$$\begin{aligned}\lim_{\sigma\to 0} E(\hat{\omega}_i^{(1)} - \omega_i^*) \\ &= E\left(0 - \omega_i^*\right) \\ &= -\omega_i^*\end{aligned}$$

Now we can rewrite as desired.

$$\lim_{\sigma\to 0} E\left(\hat{w}_i^{(1)} - w_i^*\right) = \begin{cases} -\lambda, & \text{if } w_i^* > \lambda, \\ -w_i^*, & \text{if } |w_i^*| \leq \lambda, \\ \lambda, & \text{if } w_i^* < -\lambda. \end{cases} \tag{15}$$

LASSO which stands for (Least Absolute Shrinkage and Selection Operator) reduces the risk of getting large variances for our models but from the expression found in equation (15) we can see that with large lambdas we get a high risk of having large biases which reduces the functionality of our model.

4 Task 4

The purpose of task 4 was to try and reconstruct a signal using LASSO with different values for lambda. The first two tries we set lambda to 0.1 and 10. For the last lambda value I varied it to find a proper value to solve the task optimally. After trying some different values I thought the best one setting it to 1. The signal we were supposed to reconstruct was the linear combination of two sinusoids which had different frequencies.

As we can see for lambda being set to 10 the we get and underfitted model, i.e the model does not fit the the real data points well and we get a large error. If we look at lambda equal to 0.1, we as can be seen in the graph, clearly overfit the model as it passes through all the real data points and we start fitting on noise and get a way to complex model. But for lambda being 1 we get a good fit on the model, not overfitted since it does not fit the noise part of the data. When compared to actual graph which was provided in the labnotes the similarity between the actual signal without noise and our signal looks very similar.

When tracking the amount on non-zero coordinates we can see that it is inversely proportional to the lambda value we choose to use. As the lambda value decreases the amount of non zero coordinates increase. Having a large lambda leads to fewer trainable parameters and in our case our signal only had four but as we can see from the graph using lambda equal to 10 gives six non-zero coordinates which clearly was not enough to make a good fit.

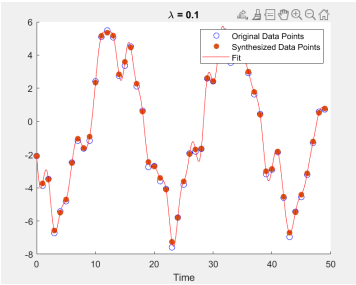


Figure 1: λ 0.1

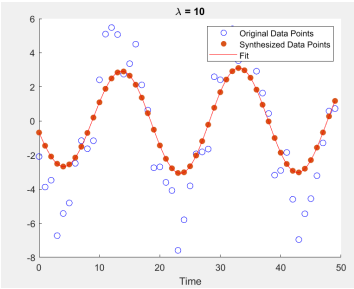


Figure 2: λ 10

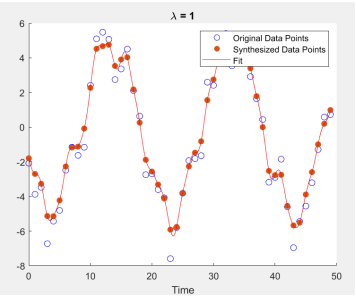


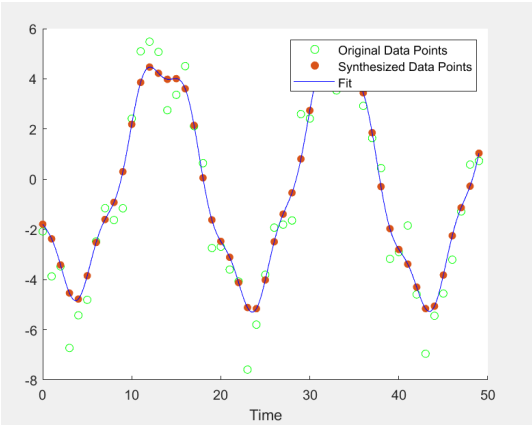
Figure 3: λ 1

λ	Non-zero coordinates
0.1	234
10	6
1	62

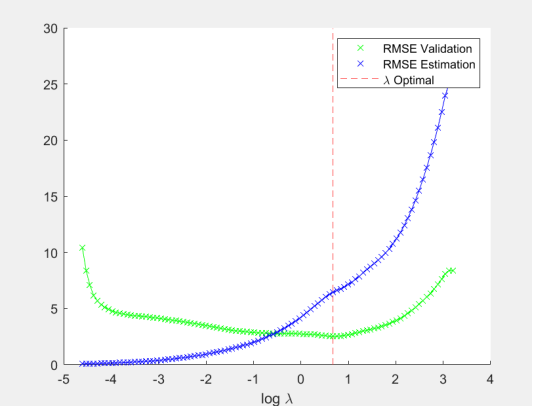
Table 1: Amount of non-zero paramaters for three different lambdas

5 Task 5

The purpose of task 5 was to find the actual optimal lambda by using 10-fold cross validation instead of guessing like in task 4. In order to do this we minimized the root mean squared error of the validation set for 100 different values of lambda ranging from 0.01 and max $\left(\left|X^T t\right|\right)$. As we can see from the results below, The training error is really small for small lambdas but not for the validation due to the overfitting during training. The optimal lambda was found to be around 1.8 which was not to far off from my guess during task 4. The reconstruction using the optimal lambda is also found below as well as the the validation and training error for different lambdas.



(a) Fit with optimal lambda



(b) Training and validation error for different values of lambda

Figure 4: Plots for task 5

6 Task 6

For me to find the optimal lambda for the entire set of audio datapoints, I used multiframe 3-fold cross-validation. In order to get a viable estimation of how well a value for it performed across all frame hence I took the mean RMSE from each lambda for all datasegments. In order to determine the optimal lambda I just choose the one which performed best across all data sets on average. Below is the graph for the RMSE for training and validation. As we can see the lambda has a clear optimal range and the actual optimal value was recorded as around 0.0044. If we were to use much a smaller or

much larger regularization value we would get noisy data. The graph showing validation and training error is presented below in figure 5.

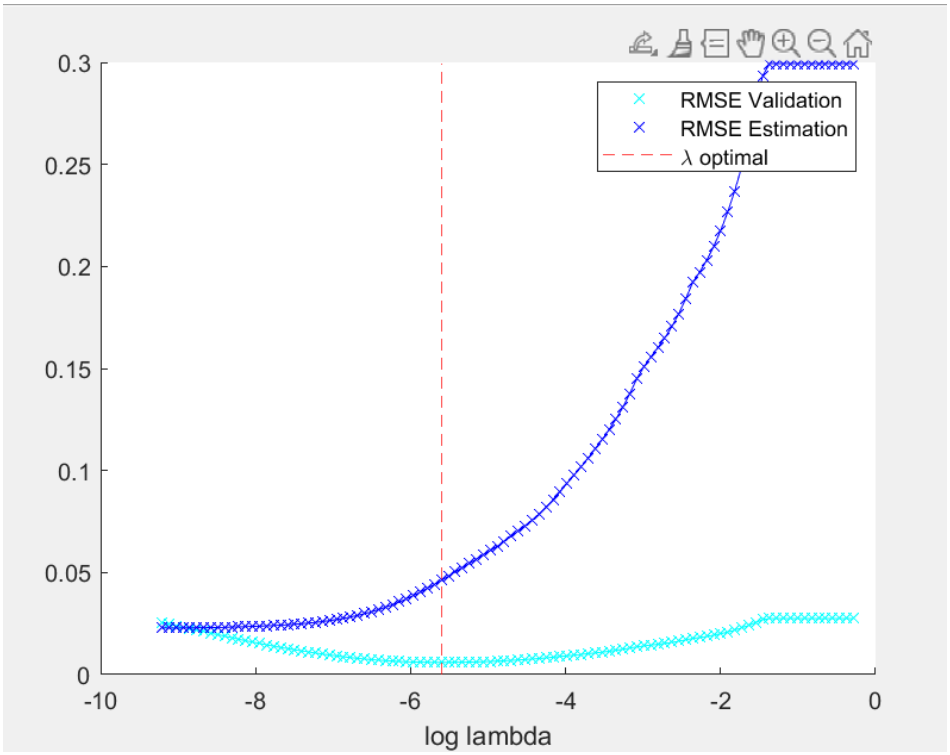


Figure 5: Finding optimal lambda for removing noise

7 Task 7

The final task was to see how well our lambda performed by listening to it. The original audio was extremely noisy and after using the optimal lambda obtained from task 6 to denoise the audio it actually sounded a lot better, not perfect at all but still much better. You could still hear a lot of noise however much less, which I must say was nice since the training took a lot of time, almost 8 hours so I don't recommend running that except if you want to recalculate the lambda.