Home Assignment 1

Arvid Gramer

FMAN45 Machine Learning
Lunds Tekniska Högskola, Lunds Universitet
April 2023

1 Penalized regression via the LASSO

Task 1: Our objective is to solve the minimisation problem

$$\arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{t} - \mathbf{X}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1 \tag{1}$$

where $\lambda \geq 0$. Since, in general, no closed form solution exists for this, we use a coordinate wise method where we iterate over each coordinate in $\mathbf{w}$ and solve

$$\arg\min_{w_i} \frac{1}{2}||\mathbf{r}_i - \mathbf{x}_i w_i||_2^2 + \lambda|w_i|. \tag{2}$$

In this equation $\mathbf{x}_i$ corresponds to the values of the i:th feature and $\mathbf{r}_i$ is $\mathbf{t} - \sum_{l \neq i} \mathbf{x}_l w_l$, namely the part of the residual that corresponds to the other features that we do not seek to change in the current step. We now seek a solution to this element wise optimisation. We call the expression Q and begin with differentiating ut with respect to w_i, resulting in:

$$\frac{\partial Q}{\partial w_i} = -\mathbf{x}_i^\intercal(\mathbf{r}_i - \mathbf{x}_i w_i) + \lambda \text{ sign}(w_i) \stackrel{!}{=} 0 \tag{3}$$

With $\mathbf{x}_i \neq \mathbf{0}$ we can rewrite this expression as:

$$w_i = \frac{\mathbf{x}_i^\intercal \mathbf{r}_i}{\mathbf{x}_i^\intercal \mathbf{x}_i} - \frac{\text{sign}(w_i)\lambda}{\mathbf{x}_i^\intercal \mathbf{x}_i} \tag{4}$$

We now have an entangled sign-operator, since sign(w_i) corresponds to the sign of the right hand side of the equation, which has w_i in it. It is now worth noting that these numbers that yield the new value of w_i are from the previous iteration over $\mathbf{w}$ and luckily, since in our case $\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)} > \lambda$ we can see that the sign of the expression will be determined by the sign of $\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}$. This yields sign (w_i) = sign $(\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)})$. Rewriting the sign-operator as sign$(\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}) = \mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}/|\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}|$ we get

$$w_i^{(j)} = \frac{\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^\intercal \mathbf{x}_i}\left(1 - \frac{\lambda}{|\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}|}\right) = \frac{\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^\intercal \mathbf{x}_i |\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}|}\left(|\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}| - \lambda\right) \tag{5}$$

and we are finished.

Task 2: Our task is to show that if the regression matrix $\mathbf{X}$ is an orthonormal basis, the coordinate descent solver converges in at most one pass over the coordinates in $\mathbf{w}$. This translates to $w_i^{(2)} - w_i^{(1)} = 0, \forall i$. To find if this holds we simply rewrite the expression for w_i:

$$w_i^{(j)} = \frac{\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^\intercal \mathbf{x}_i |\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}|}\left(|\mathbf{x}_i^\intercal \mathbf{r}_i^{(j-1)}| - \lambda\right) \tag{6}$$

Because of $\mathbf{X}$'s orthonormality we have that $\mathbf{x}_i^\intercal \mathbf{x}_i = 1$ and

$$\mathbf{x}_i^\intercal \mathbf{r}_i^{(j)} = \mathbf{x}_i^\intercal \big(\mathbf{t} - \sum_{l<i} \mathbf{x}_l \hat{w}_l^{(j)} - \sum_{l>i} \mathbf{x}_l \hat{w}_l^{(j-1)}\big) = \mathbf{x}_i^\intercal \mathbf{t} \tag{7}$$

due to $\mathbf{x}_i^\intercal \mathbf{x}_l = 0, i \neq l$. This yields

$$w_i^{(j)} = \mathbf{x}_i^\intercal \mathbf{t}\Big(1 - \frac{\lambda}{|\mathbf{x}_i^\intercal \mathbf{t}|}\Big) \tag{8}$$

which implies $w_i^{(2)} = w_i^{(1)}$ since none of the elements in the equation is dependent of which iteration it is.

Task 3: Our task is to show that the LASSO estimate's bias tends towards

$$\lim_{\sigma \to 0} \mathbb{E}[\hat{w_i}^{(1)} - w_i^*] = \begin{cases} -\lambda & w_i^* > \lambda \\ -w_i^* & |w_i^*| \leq \lambda \\ \lambda & w_i^* < \lambda \end{cases} \tag{9}$$

when the data is generated by $\mathbf{t} = \mathbf{X}\mathbf{w}^* + \mathbf{e}, \mathbf{e} \sim N(\mathbf{0}, \sigma\mathbf{I})$ and the regression matrix is orthonormal. Using the orthonormality and the result from task 2 we can rewrite equation (3) in the assignment description as

$$\hat{w_i} = \begin{cases} \mathbf{x}_i^\intercal \mathbf{t}\Big(1 - \frac{\lambda}{|\mathbf{x}_i^\intercal \mathbf{t}|}\Big) & |\mathbf{x}_i^\intercal \mathbf{t}| > \lambda \\ 0 & |\mathbf{x}_i^\intercal \mathbf{t}| \leq \lambda \end{cases} \tag{10}$$

With $\mathbf{t} = \mathbf{X}\mathbf{w}^* + \mathbf{e}$, and orthormality which yields $\mathbf{x}_i^\intercal \mathbf{X}\mathbf{w}^* = \mathbf{x}_i^\intercal \mathbf{x}_i w_i^* = w_i^*$, we can rewrite the equations as

$$\hat{w_i} = \begin{cases} (w_i^* + \mathbf{x}_i^\intercal \mathbf{e})\Big(1 - \frac{\lambda}{|w_i^* + \mathbf{x}_i^\intercal \mathbf{e}|}\Big) & |w_i^* + \mathbf{x}_i^\intercal \mathbf{e}| > \lambda \\ 0 & |w_i^* + \mathbf{x}_i^\intercal \mathbf{e}| \leq \lambda \end{cases} \tag{11}$$

As $\sigma \to 0$, the stochastic error vector $\mathbf{e}$ tends towards it's expected value $= \mathbf{0}$. This yields a new expression for the estimated weight:

$$\lim_{\sigma \to 0} \hat{w_i} = \begin{cases} w_i^*\Big(1 - \frac{\lambda}{|w_i^*|}\Big) & |w_i^*| > \lambda \\ 0 & |w_i^*| \leq \lambda \end{cases} \tag{12}$$

$\lim_{\sigma \to 0} \mathbb{E}[\hat{w_i}^{(1)} - w_i^*]$ then becomes:

$$\begin{cases} w_i^* - \lambda \text{ sign}(w_i^*) - w_i^* & |w_i^*| > \lambda \\ 0 - w_i^* & |w_i^*| \leq \lambda \end{cases} = \begin{cases} -\lambda & w_i^* > \lambda \\ w_i^* & |w_i^*| \leq \lambda \\ \lambda & w_i^* < \lambda \end{cases} \tag{13}$$

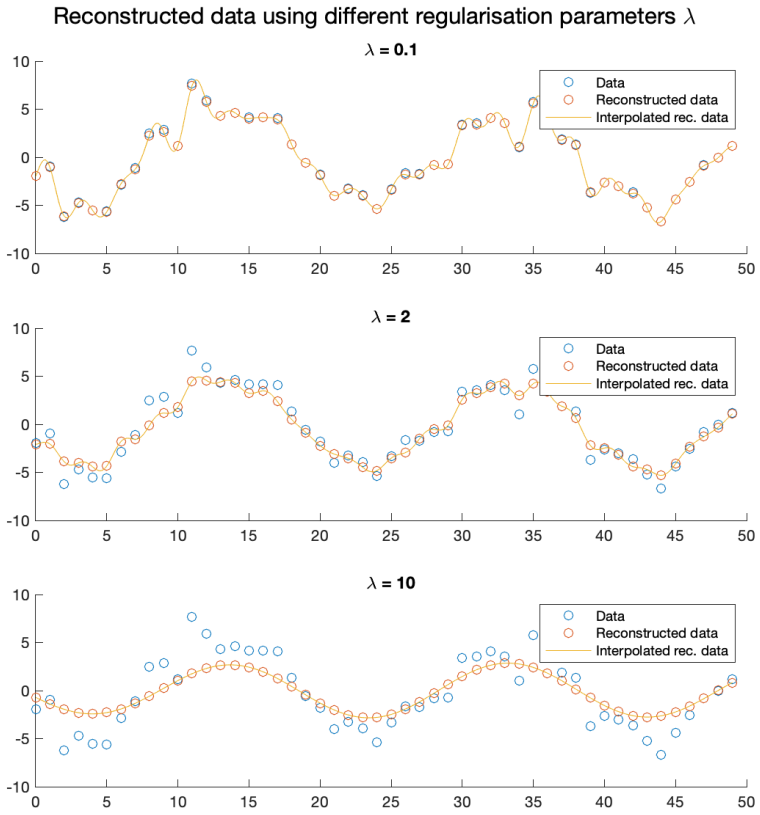

Figure 1: The data, reconstructed data and an interpolation of the reconstructed data for different choices of λ. In the top plot, with $\lambda = 0.1$ it is clear that the optimiser barely penalises the amount of frequencies included in the reconstruction. The second is the user chosen $\lambda = 2$, and shows a more balanced reconstruction, where some are accepted and some not. The last plot penalises all but two weights, creating a plain sinusoid.

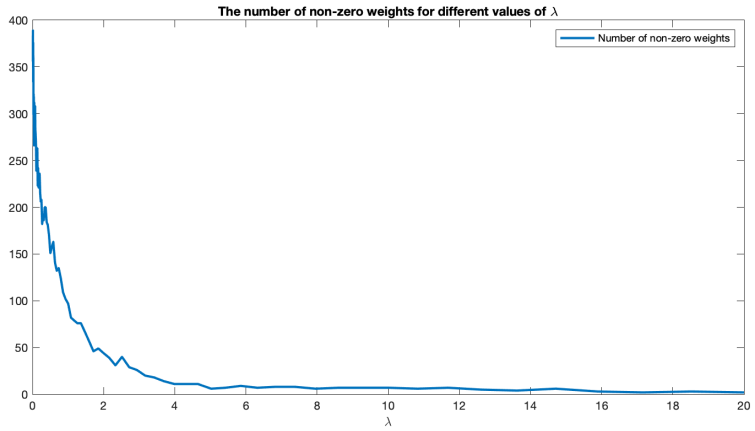

Figure 2: The number of non-zero parameters in the resulting weight vector. As λ increases, the number of parameters tend to 0, after a long plateau of a number around 2.

2 Hyperparameter-learning via cross-validation

Task 4: We begin with implementing a cyclic coordinate descent solver for the coordinate-wise LASSO and plot data as well as the reconstructed data in figure 1. In order to understand the effect of the penalty hyperparameter λ we run the optimiser over a grid of 100 different values of λ between 0.01 and 20 and count the resulting number of non-zero weights, illustrated in figure 2. For λ between 4 and 16 the number of non-zero parameters stay around 2. Above that, the penalty is too high to allow for any parameters at all.

Task 5: We continue by implementing a K-fold cross validation scheme for the LASSO solver. We use an array of 100 candidate λ:s logaritmically distributed between 0.01 and 10. This is illustrated by a plot in figure 3 of the root mean squared error from both estimation and validation data, for the different λ. The difference between these errors is a measure of the generalisation error. For low regularisation, the model performs better on the estimation data and worse on validation data. Since we are interested in the performance on unseen data we optimise using the model that performs best on validation data. The weights are then re-estimated using the optimal lambda on the full dataset. The data points reconstructed using these weights are plotted in figure 4. The number of non-zero coordinates with this value is 45, which far from the correct number (=4) so the solver obviously fits to much to the noise. However, with only 50 noisy data points and since it ocularly resembles the ground truth, we can be somewhat satisfied with the result.

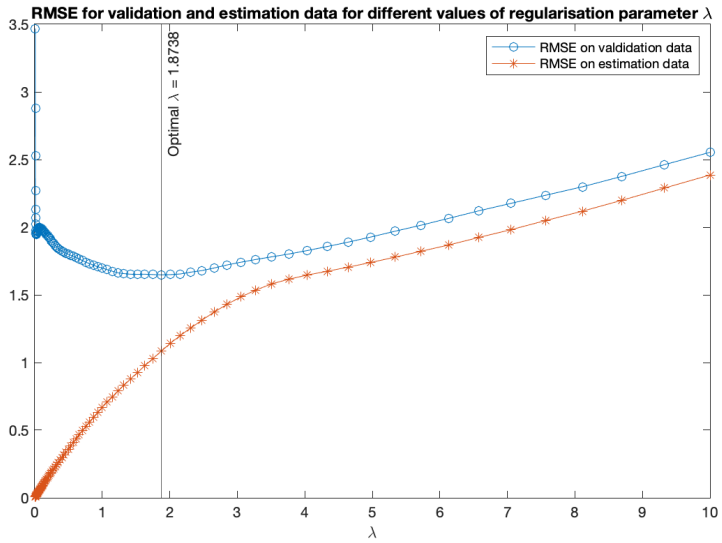

Figure 3: The root mean square error (RMSE) over a grid of logarithmically spaced values of λ between 0.01 and 10. The values are measured from a $k = 5$ fold cross validation. The optimal λ is the one that minimizes the validation error, as this (hopefully) is a proxy for unseen future data.

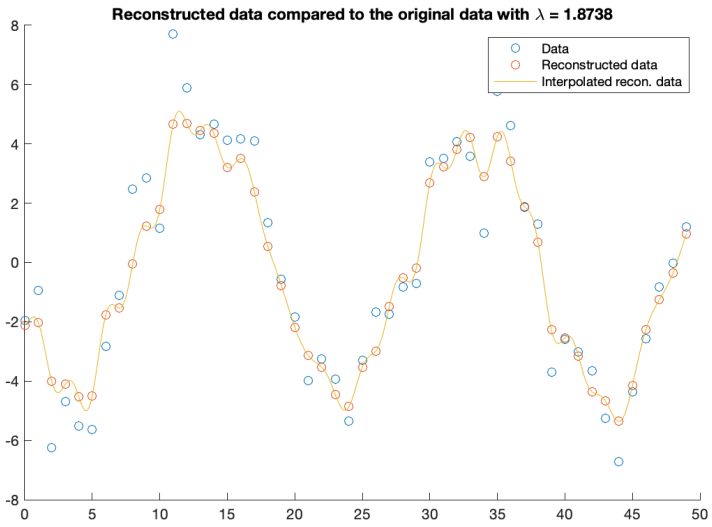

Figure 4: The reconstructed and real data when using the cross validation optimal $\lambda = 1.8738$. The number of non-zero weights for this value is 45, which is too high compared to the two pairs of frequencies = four weights.

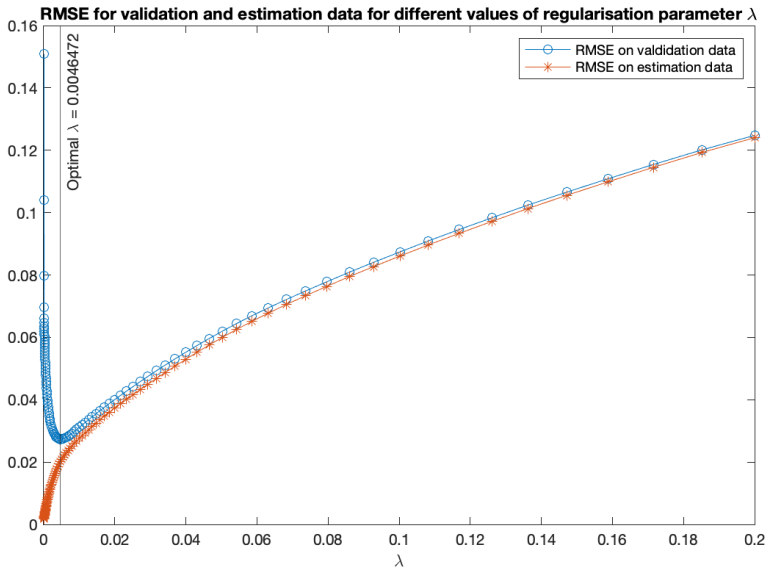

Figure 5: Cross validation over different values of λ for the multi-frame estimation of weights for denoising audio. The optimal $\lambda = 4.6 \cdot 10^{-3}$ is marked.

3 Denoising of an audio excerpt

Task 6: We implement a frame-wise cross validation scheme for LASSO optimisation in order to try do denoise an audio recording. The frames are meant to be short enough to make stationarity assumptions on the audio suffice. For this we immediately notice that the hyperparameter λ must be searched for in much smaller magnitudes than the previous assignment. The grid we search in is logarithmically spaced between 10^{-4} and 0.2. The resulting RMSE for validation and estimation data, as well as optimal λ is found in figure 5. The low value of optimal λ indicates that we should not remove too many frequencies, probably indicating that the audio still needs many frequencies to be well represented.

Task 7: We then use the optimal λ from task 6 and to denoise the audio. When we do this, some problems occur. During the denoising process, many prompts about deficient matrix ranks are presented from Matlab. This then results in a not very good denoising, where there the sound is very "wobbly" and not at all denoised. I have tried to debug this, and also varying λ in order to remove more/less features but without success.