FMAN45 Machine Learning - Assignment 1

Erik Waldemarson

1 Penalized regression via the LASSO

In this part I will derive some equations used for regression with LASSO.

1.1 Exercise 1

The coordinate-wise LASSO solves the minimization problem

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

for some $\lambda \geq 0$, where w_i is the i:th coordinate of a weight vector $\mathbf{w} \in \mathbb{R}^M$, $\mathbf{r_i} = \mathbf{t} - \sum_{l \neq i} \mathbf{x_l} w_l$ is the residual vector without the effect of the i:th regressor for some target $\mathbf{t} \in \mathbb{R}^N$ and $\mathbf{x_i}$is the i:th column of the regression matrix $X \in \mathbb{R}^{N \times M}$.

I begin by defining the function

$$H(w_i) := \frac{1}{2}\|\mathbf{r_i} - \mathbf{x_i} w_i\|_2^2 + \lambda |w_i| = \frac{1}{2}(\mathbf{r_i}^T \mathbf{r_i} - 2\mathbf{x_i}^T \mathbf{r_i} w_i + \mathbf{x_i}^T \mathbf{x_i} w_i^2) - \lambda |w_i|. \tag{2}$$

Solving (1) is equivalent to minimizing $H(w_i)$, which is guaranteed to have a solution since it's a convex function (since it's a non-negative weighted sum of norms which is always convex). I this do by finding stationary points for non-zero w_i:

$$\frac{dH}{dw_i} = -\mathbf{x_i}^T \mathbf{r_i} w_i + \mathbf{x_i}^T \mathbf{x_i} w_i - \lambda w_i / |w_i| = 0, \quad w_i \neq 0. \tag{3}$$

Rearranging (3) yields

$$w_i = \frac{\mathbf{x_i}^T \mathbf{r_i} - \lambda w_i / |w_i|}{\mathbf{x_i}^T \mathbf{x_i}}, \tag{4}$$

which can be split up into two cases:

Case 1: $w_i > 0 <=> w_i = |w_i|$

$$=> |w_i| = \frac{\mathbf{x_i}^T \mathbf{r_i} - \lambda}{\mathbf{x_i}^T \mathbf{x_i}}.$$

Case 2: $w_i < 0 <=> w_i = -|w_i|$

$$=> |w_i| = \frac{-\mathbf{x_i}^T\mathbf{r_i} - \lambda}{\mathbf{x_i}^T\mathbf{x_i}}.$$

Both cases can be written as

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

assuming $|\mathbf{x_i}^T\mathbf{r_i}| - \lambda > 0$. Otherwise the only solution is $|w_i| = 0 => w_i = 0$ (since $\lambda \geq 0$) which contradicts the assumption $w_i \neq 0$.

Rearranging (4) and inserting (5) yields

$$w_i = \frac{\mathbf{x_i}^T\mathbf{r_i}|w_i|}{\lambda + \mathbf{x_i}^T\mathbf{x_i}|w_i|} = \frac{\mathbf{x_i}^T\mathbf{r_i}}{\mathbf{x_i}^T\mathbf{x_i}|\mathbf{x_i}^T\mathbf{r_i}|}(|\mathbf{x_i}^T\mathbf{r_i}| - \lambda), \quad |\mathbf{x_i}^T\mathbf{r_i}| > 0. \tag{6}$$

Now since each update is done iteratively I have to define $\hat{w}_i^{(j)}$ for some iteration j. Assuming this is done cyclically i.e. in order, at each iteration j I have access to all the update previous weights $\hat{w}_l^{(j)}$ for $l < i$ but for upcoming weights I only have access to the old ones, $\hat{w}_l^{(j-1)}$ for $l > i$. So the best regression vector $\mathbf{r_i}$ I can use is the effect of the updated previous regressors and the upcoming old regressors, call this $\mathbf{r_i}^{(j-1)}$. So by making these changes to (6) I get in total

$$\hat{w}_i^{(j)} = \frac{\mathbf{x_i}^T\mathbf{r_i}^{(j-1)}}{\mathbf{x_i}^T\mathbf{x_i}|\mathbf{x_i}^T\mathbf{r_i}^{(j-1)}|}(|\mathbf{x_i}^T\mathbf{r_i}^{(j-1)}| - \lambda), \quad |\mathbf{x_i}^T\mathbf{r_i}^{(j-1)}| > 0, \tag{7}$$

where $\mathbf{r_i}^{(j-1)} = \mathbf{t} - \sum_{l<i}\mathbf{x_l}\hat{w}_l^{(j)} - \sum_{l>i}\mathbf{x_l}\hat{w}_l^{(j-1)}$. □

1.2 Exercise 2

I will show that if the regression matrix X is orthonormal then weights do not depend on the iteration, i.e. $\hat{w}_i^{(1)} = \hat{w}_i^{(2)} = \hat{w}_i$. An orthonormal matrix has the property

$$\mathbf{X}^T\mathbf{X} = I_N <=> \mathbf{x_i}^T\mathbf{x_j} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$$

If X is orthonormal then I have the simplification

$$\mathbf{x_i}^T\mathbf{r_i}^{(j-1)} = \mathbf{x_i}^T\mathbf{t} - \sum_{l<i}\mathbf{x_i}^T\mathbf{x_l}\hat{w}_l{}^{(j)} - \sum_{l>i}\mathbf{x_i}^T\mathbf{x_l}\hat{w}_l{}^{(j)} = \mathbf{x_i}^T\mathbf{t} - 0 - 0 = \mathbf{x_i}^T\mathbf{t}. \tag{8}$$

Inserting (8) into (7) with an extra cases gives

$$\hat{w}_i{}^{(j)} = \hat{w}_i = \begin{cases} \frac{\mathbf{x_i}^T\mathbf{t}}{|\mathbf{x_i}^T\mathbf{t}|}(|\mathbf{x_i}^T\mathbf{t}| - \lambda), & |\mathbf{x_i}^T\mathbf{t}| > \lambda, \\ 0, & |\mathbf{x_i}^T\mathbf{t}| \leq \lambda, \end{cases} \tag{9}$$

which shows that $\hat{w}_i$ doesn't depend j. □

1.3 Exercise 3

First notice that there are two cases:
Case 1: $|\mathbf{x_i}^T\mathbf{t}| = \mathbf{x_i}^T\mathbf{t} => \mathbf{x_i}^T\mathbf{t} > -\lambda$
Case 2: $|\mathbf{x_i}^T\mathbf{t}| = -\mathbf{x_i}^T\mathbf{t} => \mathbf{x_i}^T\mathbf{t} < -\lambda$.
So I can split up (9) into three cases

$$\hat{w_i}^{(1)} = \hat{w_i} = \begin{cases} \mathbf{x_i}^T\mathbf{t} - \lambda, & \mathbf{x_i}^T\mathbf{t} > \lambda, \\ 0, & |\mathbf{x_i}^T\mathbf{t}| \leq \lambda, \\ \mathbf{x_i}^T\mathbf{t} + \lambda, & \mathbf{x_i}^T\mathbf{t} < -\lambda. \end{cases} \tag{10}$$

I can further calculate that

$$\mathbf{x_i}^T\mathbf{t} = \mathbf{x_i}^T\mathbf{X}\mathbf{w}^* + \mathbf{x_i}^T\mathbf{e} = w_i^* + \mathbf{x_i}^T\mathbf{e}, \tag{11}$$

since $\mathbf{X}$ is an orthonormal matrix.

Using the facts that $E(aX + b) = aE(X) + b$ and $E(\mathbf{e}) = \mathbf{0}_N$ I get

$$E(\hat{w_i}^{(1)} - w_i*) = \begin{cases} w_i^* + \mathbf{x_i}^T E(\mathbf{e}) - \lambda - w_i^* = -\lambda, & w_i^* + \mathbf{x_i}^T\mathbf{e} > \lambda, \\ 0 - w_i^* = -w_i^*, & |w_i^* + \mathbf{x_i}^T\mathbf{e}| \leq \lambda, \\ w_i^* + \mathbf{x_i}^T E(\mathbf{e}) + \lambda - w_i^* = \lambda, & w_i^* + \mathbf{x_i}^T\mathbf{e} < -\lambda. \end{cases} \tag{12}$$

Now it's clear in the limit that

$$\lim_{\sigma \to 0} w_i^* + \mathbf{x_i}^T\mathbf{e} = w_i^* \tag{13}$$

since the normal distribution will have zero standard deviation and thus $\mathbf{e} \to \mathbf{0}_N$ as $\sigma \to 0$. Finally I get

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

□

This is related to the acronym LASSO (Least Absolute Shrinkage and Selection Operator) since there will be some bias bounded in the interval $[-\lambda, \lambda]$ depending on the size of the weight w_i^*. So larger weights will be punished in some sense.

2 Hyperparameter-learning via K-fold cross-validation

Here I will use K-fold cross-validation to find a suitable hyperparameter λ.

2.1 Exercise 4

I implemented the function lasso_ccd() and used it to find $\mathbf{w}$ for three different $\lambda = [0.1, 0.8, 10]$ which have been plotted in Figures 1-3.

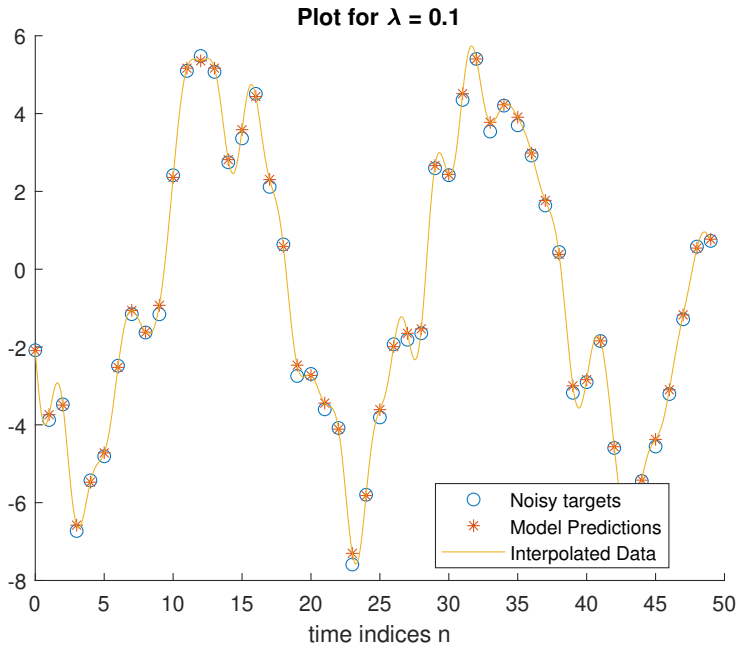

Figure 1: Noisy targets, model predictions and interpolated reconstruction of the data for $\lambda = 0.1$.

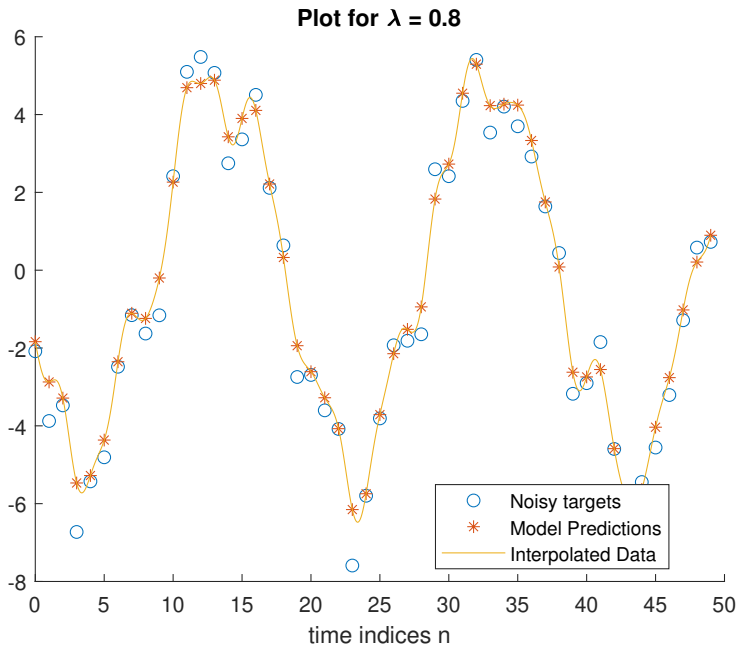

Figure 2: Noisy targets, model predictions and interpolated reconstruction of the data for $\lambda = 0.8$.

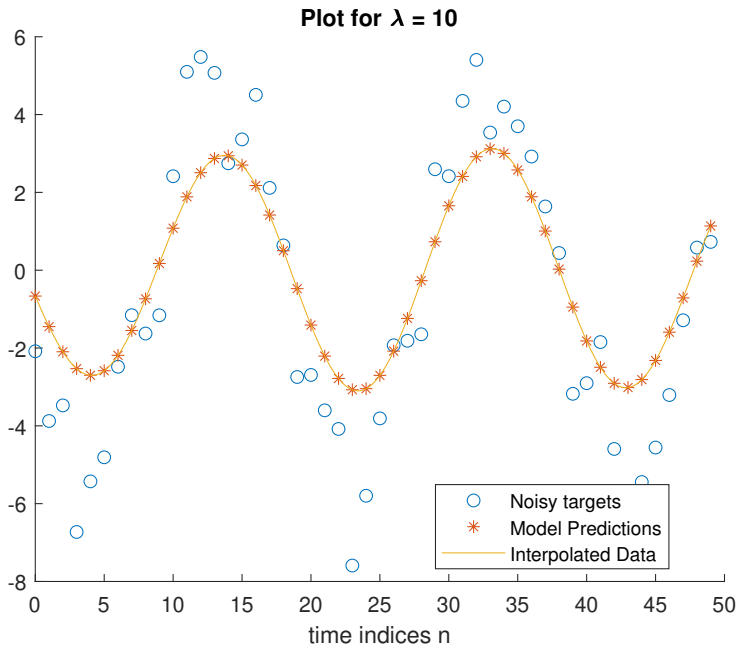

Figure 3: Noisy targets, model predictions and interpolated reconstruction of the data for $\lambda = 10$.

As one can see in Figures 1 and 3, a lower λ will lead to more overfitting to noise while a very high λ will lead to high generalization but fits poorly to the data. I choose $\lambda_{\text{user}} = 0.8$ because it looked like a good trade-off between generalization and fitting to data, but it's hard to tell just from looking at it.

For each λ I get the following number of non-zero coordinates of $\mathbf{w}$:

$\lambda = 0.1$: 252,

$\lambda = 0.8$: 76,

$\lambda = 10$: 6.

As you can see, it's very difficult for it to reach the true number of non-zero w_i needed even with high λ.

2.2 Exercise 5

I implemented the K-fold cross validation outlined in instructions in the function lasso_cv(). The validation error $RMSE_{\text{val}}(\lambda_j)$ and estimation error $RMSE_{\text{est}}(\lambda_j)$ have been plotted against different λ in Figure 4 when I split the data $K = 5$ subsets.

As expected, the estimation error increases with λ as it becomes harder for the model to make a good regression as more w_i are forced to 0. The validation error has a minimum due to the bias-variance tradeoff: Error = bias^2 + var +

σ. Simpler models (higher λ) will have higher bias since they can't fit well to the data, however they will have lower variance for different datasets since it has less overfitting. The measurement noise is given by σ.

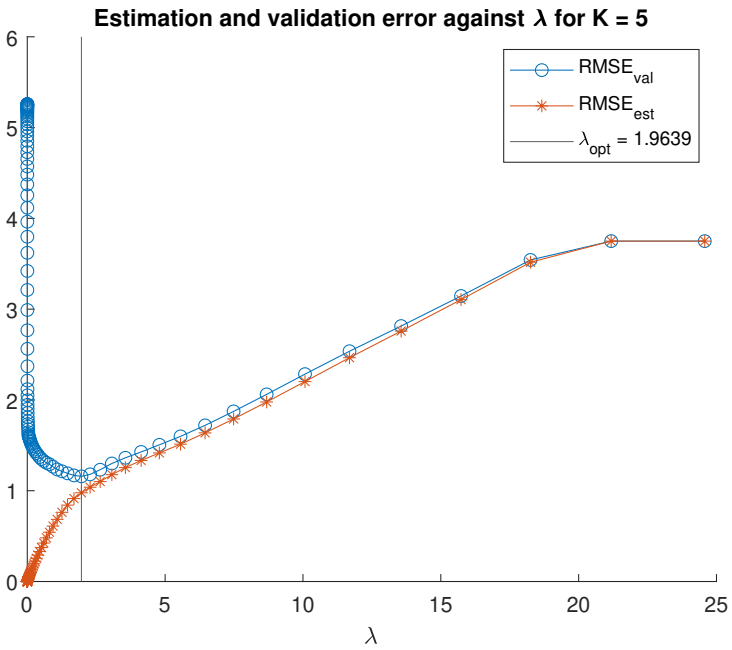

Figure 4: The validation error $RMSE_{\text{val}}(\lambda_j)$ and $RMSE_{\text{est}}(\lambda_j)$ with K-cross validation with $K = 5$ for different λ together with λ_{opt}.

The optimal λ_{opt} was found to be $\lambda_{\text{opt}} = 1.9639$. The data was reconstructed with LASSO-regression as in the previous section with $\lambda = \lambda_{\text{opt}} = 1.9639$ and is plotted in Figure 5.

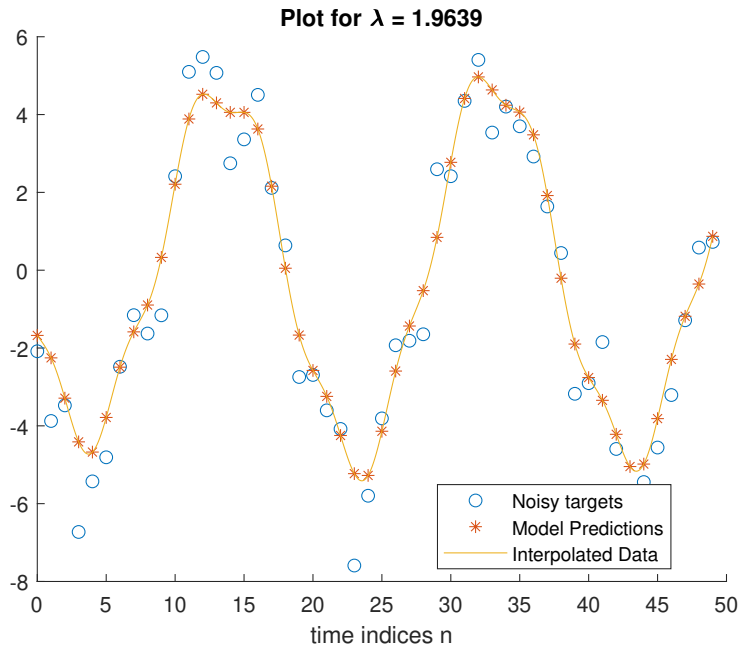

Figure 5: Noisy targets, model predictions and interpolated reconstruction of the data for $\lambda = \lambda_{\text{opt}} = 1.9639$.

3 Denoising of an audio excerpt

Here I use K-fold LASSO to denoise an audio excerpt.

3.1 Exercise 6

This time I do the same thing as in the previous section, that is K-fold cross-validation with LASSO, but the data is now an audio file that has to be split up into multiple frames. The optimal λ_{opt} is then found by calculating the RMSE error over all frames.

I implement the function multiframe_lasso_cv() and perform the K-fold regression with $K = 5$ once more. The error together with $\lambda_{\text{opt}} = 0.044316$ for K $= 5$ has been plotted in Figure 6. I accidentally used a very high λ_{max} which can be seen in the graph but I decided to ahead with these results anyway.

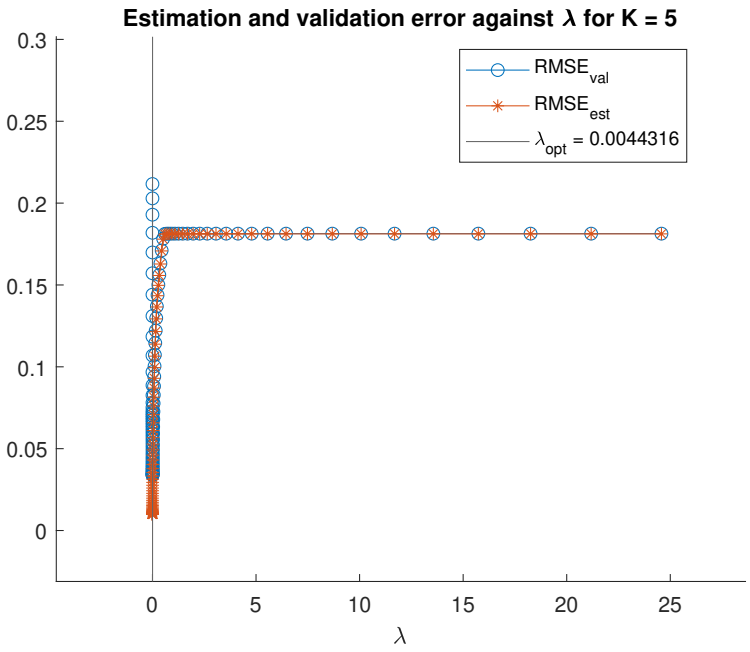

Figure 6: The validation error $RMSE_{\text{val}}(\lambda_j)$ and $RMSE_{\text{est}}(\lambda_j)$ with multiframe K-cross validation with $K = 5$ for different λ together with λ_{opt}.

It seems that a very low λ is good for this task.

3.2 Exercise 7

I then denoised the Ttest audio using the lasso_denoise() function with the optimal $\lambda_{\text{opt}} = 0.044316$. I listened to it but it still sounded very noisy to me, so I found another $\lambda = 0.02$ which I thought was better. It may prove to be worse at generalization though or perhaps different K needs to be tried out in the K-cross validation to get a good λ.