# Machine Learning, LTH 2021

*Assignment 1: Penalized Regression and Cross-validation*

Name: Isabelle Frodé
Date: April 5 2021

# 1 Assignment Overview

This assignment was given in the course Machine Learning FMAN45 at LTH 2021. The goal with the assignment was to derive and implement a penalized regression algorithm using *Least Absolute Shrinkage and Selection Operatior* (LASSO). K-fold cross-validation was then used to determine the optimal value of the hyperparameter that controls the impact of different weights. for example by setting non-necessary weights to zero. The LASSO and K-fold cross-validation was then extended to function for multiframe data and then applied to a short audio excerpt of piano music to perform noise cancelling. The result show that the implemented algorithms will reduce background noise in some extend, but does not manage to perform perfect noise cancelling.

# 2 Penalized Regression via the LASSO

When performing linear regression we start with an optimization goal declared in the objective function. To control the capacity a regularization term $\lambda$ that look at the weights and penalize large weights can be added. The goal with penalized regression is that even though the regression line will fit the training data worse by generating a higher bias, hopefully it will fit the test data better, providing low variance. The LASSO in particularly is commonly used when there is few non-zero coordinates and it solves the following minimization problem:

$$\underset{\mathbf{w}}{\text{minimize}} \frac{1}{2}||\mathbf{t} - \mathbf{X}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1 \tag{1}$$

where $\mathbf{w}$ is the weights, $\mathbf{t}$ the outcome variable (the data), $\mathbf{X}$ the regression matrix and $\lambda$ the capacity hyperparameter.

## 2.1 Derivation of closed form solution for the coordinate descent minimizer

There is no closed form solution to the LASSO optimization problem, but the convex properties allows for a convergent coordinate step-wise approach. The method will update the weights, one weight at the time, with the other weights fixed. The coordinate descent minimization problem for each step can be formulated:

$$\underset{w_i}{\text{minimize}} \frac{1}{2}||\mathbf{r}_i - \mathbf{x}_i w_i||_2^2 + \lambda|w_i| \tag{2}$$

where $i$ is the index for the i:th vector in the regression matrix $\mathbf{X}$ and the residual vector, $\mathbf{r}_i = \mathbf{t} - \sum_{l \neq i} \mathbf{x}_l w_l$.

Equation 2 can be solved by expanding the first term:

$$\frac{1}{2}||\mathbf{r}_i - \mathbf{x}_i w_i||_2^2 = \frac{1}{2}\mathbf{r}_i^T\mathbf{r}_i - \mathbf{x}_i^T\mathbf{r}_i w_i + \frac{1}{2}\mathbf{x}_i^T\mathbf{x}_i w_i^2 \tag{3}$$

Since $\frac{1}{2}\mathbf{r}_i^T\mathbf{r}_i$ does not contain any variables of interest, the term is discarded. Inserting the expansion back in equation 2 gives the following expression:

$$\min_{w_i} \sum_{i=1}^{n} -\mathbf{x}_i^T\mathbf{r}_i w_i + \frac{1}{2}\mathbf{x}_i^T\mathbf{x}_i w_i^2 + \lambda|w_i| \tag{4}$$

Fixing $i$ and differentiating with respect to $\mathbf{w}_i$ gives the following:

$$\frac{d}{dw_i}(-\mathbf{x}_i^T\mathbf{r}_i w_i + \frac{1}{2}\mathbf{x}_i^T\mathbf{x}_i w_i^2 + \lambda|w_i|) = -\mathbf{x}_i^T\mathbf{r}_i + \mathbf{x}_i^T\mathbf{x}_i w_i + \frac{w_i}{|w_i|}\lambda \tag{5}$$

Setting equal to 0 to minimize:

$$w_i = \frac{\mathbf{x}_i^T\mathbf{r}_i - \frac{w_i}{|w_i|}\lambda}{\mathbf{x}_i^T\mathbf{x}_i} = \frac{\mathbf{x}_i^T\mathbf{r}_i - \text{sign}(w_i)\lambda}{\mathbf{x}_i^T\mathbf{x}_i} \tag{6}$$

Since $\mathbf{x}_i^T\mathbf{x}_i > 0$ and $\lambda \geq 0$ the $w_i$ sign function is equal to $\frac{\mathbf{x}_i^T\mathbf{r}_i}{|\mathbf{x}_i^T\mathbf{r}_i|}$ for $|\mathbf{x}_i^T\mathbf{r}_i| > \lambda$. Equation 6 can then be written:

$$w_i = \frac{\mathbf{x}_i^T\mathbf{r}_i - \frac{\mathbf{x}_i^T\mathbf{r}_i}{|\mathbf{x}_i^T\mathbf{r}_i|}\lambda}{\mathbf{x}_i^T\mathbf{x}_i} = \frac{\mathbf{x}_i^T\mathbf{r}_i}{\mathbf{x}_i^T\mathbf{x}_i|\mathbf{x}_i^T\mathbf{r}_i|}(|\mathbf{x}_i^T\mathbf{r}_i| - \lambda) \tag{7}$$

Adding the iteration $j$ and the case for $|\mathbf{x}_i^T\mathbf{r}_i| \leq \lambda$ from the assignment description [1] without derivation gives the expression for $\hat{w}_i$

$$\hat{w}_i^{(j)} = \begin{cases} \frac{\mathbf{x}_i^T\mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^T\mathbf{x}_i|\mathbf{x}_i^T\mathbf{r}_i^{(j-1)}|}(|\mathbf{x}_i^T\mathbf{r}_i| - \lambda), & |\mathbf{x}_i^T\mathbf{r}_i^{(j-1)}| > \lambda \\ 0, & |\mathbf{x}_i^T\mathbf{r}_i^{(j-1)}| \leq \lambda \end{cases} \tag{8}$$

where the residual vector $\mathbf{r}_i$ is

$$\mathbf{r}_i^{(j-1)} = \mathbf{t} - \sum_{l<i} \mathbf{x}_l\hat{w}_l^{(j)} - \sum_{l>i} \mathbf{x}_l\hat{w}_l^{(j-1)} \tag{9}$$

## 2.2 Showing that the coordinate descent solver will converge

For the simplified case where the regression matrix $\mathbf{X}$ is an orthonormal basis, the coordinate descent solver in equation 8 will converge in every step. It means that for all $i$, it holds that

$$\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = 0 \tag{10}$$

The orthonormal property of the $\mathbf{X}$ matrix implicate that $\mathbf{X}^T\mathbf{X} = \mathbf{I}_N$, $\mathbf{I}_N$ being the identity matrix. Starting from equation 8 and expanding $\mathbf{x}_i^T\mathbf{r}_i^{(j-1)}$ by inserting equation 9 gives the following expression

$$\mathbf{x}_i^T\mathbf{r}_i^{(j-1)} = \mathbf{x}_i^T(\mathbf{t} - \sum_{l<i}\mathbf{x}_l\hat{w}_l^{(j)} - \sum_{l>i}\mathbf{x}_l\hat{w}_l^{(j-1)}) \tag{11}$$

It follows from the orthonormal properties of the regression matrix that $\mathbf{x}_i^T\sum_{l<i}\mathbf{x}_l\hat{w}_l^{(j)} = 0$ and $\mathbf{x}_i^T\sum_{l>i}\mathbf{x}_l\hat{w}_l^{(j-1)} = 0$, hence

$$\mathbf{x}_i^T\mathbf{r}_i^{(j-1)} = \mathbf{x}_i^T\mathbf{t} \tag{12}$$

Inserting equation 12 in equation 8 gives the following expression for $\hat{w}_i^{(j)}$

$$\hat{w}_i^{(j)} = \frac{\mathbf{x}_i^T\mathbf{t}}{\mathbf{x}_i^T\mathbf{x}_i|\mathbf{x}_i^T\mathbf{t}|}(|\mathbf{x}_i^T\mathbf{t}| - \lambda) \tag{13}$$

Equation 13 show that $\hat{w}_i^{(j)}$ is fully independent of the iteration j and previously estimates of $\hat{w}$. The convergence of a step for the coordinate descent solver can then be proven by showing, for all $i$ it holds that

$$\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = \frac{\mathbf{x}_i^T\mathbf{t}}{\mathbf{x}_i^T\mathbf{x}_i|\mathbf{x}_i^T\mathbf{t}|}(|\mathbf{x}_i^T\mathbf{t}| - \lambda) - \frac{\mathbf{x}_i^T\mathbf{t}}{\mathbf{x}_i^T\mathbf{x}_i|\mathbf{x}_i^T\mathbf{t}|}(|\mathbf{x}_i^T\mathbf{t}| - \lambda) = 0 \tag{14}$$

## 2.3 LASSO estimate's bias

When weights are estimated using the LASSO it will be bias in the estimations, $E(\hat{\mathbf{w}} - \mathbf{w}^*) \neq \mathbf{0}$ where $\mathbf{w}^*$ is the "true" $\mathbf{w}$, the weights that defines the model that generates the data. Assuming that the data $\mathbf{t}$ is a noisy response variable defined by the regression matrix $\mathbf{X}$, it can be formulated

$$\mathbf{t} = \mathbf{X}\mathbf{w}^* + \mathbf{e}, \;\; \mathbf{e} \sim (\mathbf{0}_N, \sigma\mathbf{I}_N) \tag{15}$$

where $\mathbf{e}$ is the residuals, normally distributed around zero. The LASSO estimate's bias $E(\hat{\mathbf{w}} - \mathbf{w}^*)$ for the case where the variance $\sigma$ goes to zero can be derived from equation 8. For the simplified case where the regression

matrix has orthonormal properties, $\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}$ can be expanded using equation 12 and 15:

$$\mathbf{x}_i^T \mathbf{r}_i^{(j-1)} = \mathbf{x}_i^T \mathbf{t} = \mathbf{x}_i^T (\mathbf{x}_i w_i^* + e_i) = w_i^* + \mathbf{x}_i^T e_i \tag{16}$$

Starting from equation 8 with the case $\mathbf{x}_i^T \mathbf{r}_i^{(j-1)} > \lambda$ using that the regression matrix is an orthonormal basis we get

$$\hat{w}_i^{(j)} = \frac{\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^T \mathbf{x}_i |\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}|} (|\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}| - \lambda) = \mathbf{x}_i^T \mathbf{r}_i^{(j-1)} - \lambda \tag{17}$$

Inserting equation 16 in 17 gives

$$\hat{w}_i^{(j)} = w_i^* + \mathbf{x}_i^T e_i - \lambda \implies \hat{w}_i^{(j)} - w_i^* = \mathbf{x}_i^T e_i - \lambda \tag{18}$$

Letting the standard deviation for the distribution of the residuals go to zero implies that the residual is zero, i.e. $e_i = 0$. The estimate's bias for this case can then be written

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = -\lambda \tag{19}$$

true for the case (using equation 16)

$$\mathbf{x}_i^T \mathbf{r}_i > \lambda \;\Leftrightarrow\; w_i^* + \mathbf{x}_i^T e_i|_{e_i=0} > \lambda \;\Leftrightarrow\; w_i^* > \lambda \tag{20}$$

Starting from equation 8 with the case $|\mathbf{x}_i^T \mathbf{r}_i^{(j-1)}| \leq \lambda$ instead we need to expand both sides with the term $-w_i^*$:

$$\hat{w}_i^{(j)} = 0 \;\Leftrightarrow\; \hat{w}_i^{(j)} - w_i^* = -w_i^* \tag{21}$$

Letting the standard deviation for the residual distribution go to zero gives the estimate's bias:

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = -w_i^* \tag{22}$$

true for the case (using equation 16)

$$|\mathbf{x}_i^T \mathbf{r}_i| \leq \lambda \;\Leftrightarrow\; |w_i^* + \mathbf{x}_i^T e_i|_{e_i=0} \leq \lambda \;\Leftrightarrow\; |w_i^*| \leq \lambda \tag{23}$$

The case $\mathbf{x}_i^T \mathbf{r}_i^{(j-1)} < -\lambda$ can be derived in the same way as $\mathbf{x}_i^T \mathbf{r}_i^{(j-1)} > \lambda$. The estimate's bias for the case when $\sigma \to 0$ for all $i$ can then be written

$$\lim_{\sigma \to 0} E(\hat{w}_i^{(1)} - w_i^*) = \begin{cases} -\lambda, & w_i^* > \lambda \\ -w_i^*, & |w_i^*| \leq \lambda \\ \lambda, & w_i^* < -\lambda \end{cases} \quad \forall i \tag{24}$$

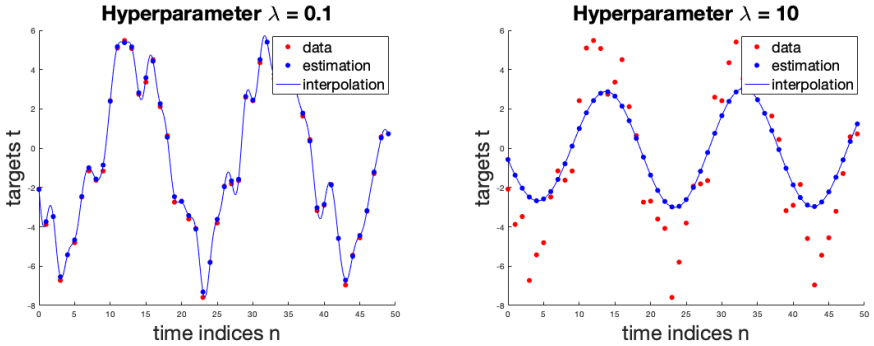
Least Absolute Shrinkage and Selection Operator (LASSO) selects the most important weights for predicting the outcome and shrinks the weights that are the least important predictor variables, leading to a higher bias. The *least absolute shrinkage* relates to that the bias will never be larger than $\lambda$. If it was no limit, it could shrink limitless and the loss could be huge. Equation 24 show that when the response variable noise goes to zero, the bias (how far away the estimated values are from the actual values) will be limited for larger weights that defines the model that generates the data.

# 3 Hyperparameter-learning via K-fold cross-validation

## 3.1 Cyclic coordinate descent solver for LASSO solution

To perform linear regression using the data $\mathbf{t}$ with regression matrix $\mathbf{X}$ given in the assignment [1], a coordinate descent solver was implemented. The solver performs cyclic solving using the derived coordinate-wise LASSO solution in equation 8.

The regression was performed using different values for the hyperparameter $\lambda$. Figure 1 show N=50 original data points along with N=50 estimated data points. The figure show two extreme cases, where the choice of a too small $\lambda$ cleary overfit the data and the choice of a too large $\lambda$ lead to underfitting. Figure 2 show the reconstructuon plot using a more suitable value of $\lambda$.




**(a)** Overfit, $\lambda = 0.1$ **(b)** Underfit, $\lambda = 10$

**Figure 1:** Visualization of the actual data, estimations and interpolation

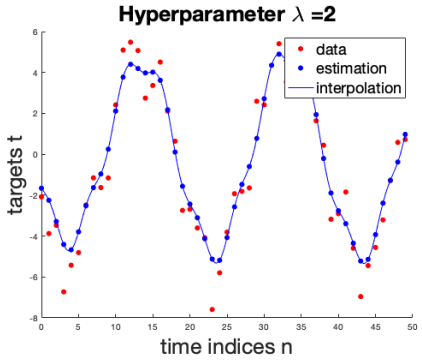



**Figure 2:** Visualization of the actual data, estimations and interpolation using a suitable value on $\lambda$

The actual number of non-zero coordinates that is required to model the data is 4 coordinates, given the frequencies 1/20 and 1/5 for N=50 data points. The number of non-zero coordinates to model the data for different values of hyperparameter $\lambda$ is shown in Table 1.

| $\lambda$ | non-zero coordinates |
|---|---|
| 0.1 | 231 |
| 2 | 17 |
| 10 | 7 |

**Table 1:** Number of non-zero coordinates for different values of hyperparameter $\lambda$

All studied $\lambda$ presented in Table 1 fit the requirement of *number of non-zero coordinates* $> 4$.

### 3.2 K-fold cross-validation scheme

To find the optimal $\lambda$ for the implemented coordinate descent LASSO solver, a *K-fold cross-validation scheme* was implemented. The cross-validation algorithm used can be found in the Assignment instructions.

For each potential value of $\lambda$ the training data is split into K number of folds. K is set to 10. K-1 folds are used for estimation and the last fold is used for validation. To evaluate the model for a specific $\lambda$ the root mean square error $RMSE_{val}(\lambda_j)$ for the validation is calculated. The optimal $\hat{\lambda}$ is chosen using

$$p = \underset{j}{\operatorname{argmin}} \; RMSE_{val}(\lambda_j) \tag{25}$$

where $\hat{\lambda} = \lambda_p$. To show the model's ability to adapt to previously unseen

data for different values of $\lambda$, the root mean square error for the estimation is calculated $RMSE_{est}(\lambda)$ as well as the root mean squared error for the validation $RMSE_{val}(\lambda)$. The result is shown in Figure 3.

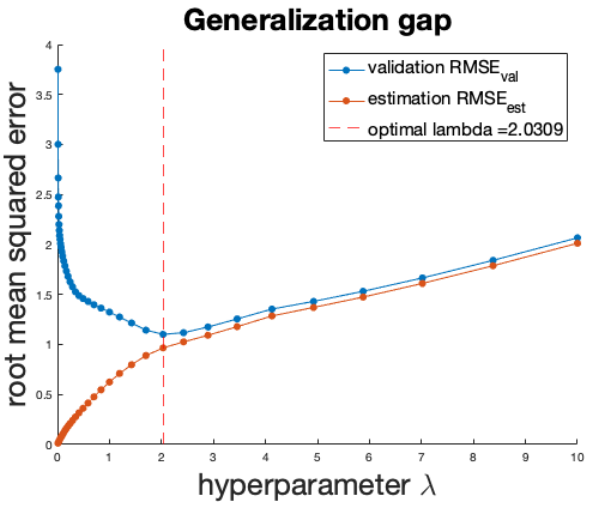



**Figure 3:** RMSE for validation and estimation for every hyperparameter $\lambda$ showing the generalization gap and selected $\lambda$

Figure 3 show that for small $\lambda$, the root mean square error is small for the estimation but large for validation. The discrepancy is what related to the overfitting shown in Figure 1. The figure further show that the optimal $\lambda$ is the minimum value of root mean squared error for the validation, which is clearly related to a small generalization gap, meaning the model will be able to adapt to previously unseen data. After the optimal $\lambda > 2$, the root mean squared error will increase and eventually underfit the data as shown in Figure 1. Figure 3 show that the model however will perform the same for training and validation data.

Figure 4 show a reconstruction plot with the hyperparameter $\lambda$ that the K-fold cross-validation scheme select as the optimal.

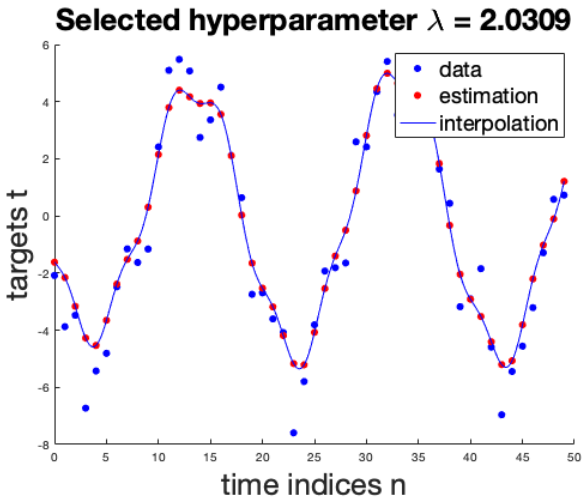



**Figure 4:** Visualization of the actual data, estimations and interpolation using $\hat{\lambda}$, the selected optimal value

# 4 Denoising of an audio excerpt

In this part, the LASSO implemented in previous sections is used to perform denoising an excerpt of piano music.

## 4.1 K-fold cross-validation scheme for multi-frame audio excerpt

The assignment contains data with 5 seconds of piano music. The data is split into a training and a test set. The training data of audio excerpts is divided into smaller parts into several "frames". One frame is 40 ms long. The K-fold cross-validation scheme was then reconstructed to find the optimal $\lambda$ for each frame by once again using the coordinate descent LASSO solver. The optimal $\lambda$ was then used to perform regression to denoise the data.

The results of the K-fold cross-validation is shown in Figure 5. The figure show the model's ability to adapt to previously unseen data by plotting the root squared error for the validation and estimation as a function of $\lambda$.

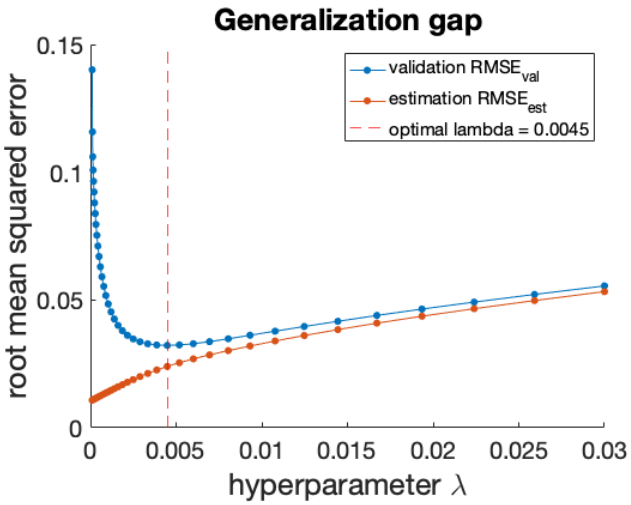



**Figure 5:** RMSE for validation and estimation for every hyperparameter $\lambda$ showing the generalization gap and selected $\lambda$

Figure 5 show that the optimal value of hyperparameter $\hat{\lambda}$ for all frames is $\hat{\lambda} = 0.0045$. The visualized generalization gap show how well the model, using different values of $\lambda$, adapt to new previously unseen data. For small values of $\lambda$ the model performs badly because of overfitting. For larger values of $\lambda$ the root mean squared error increases, but not as much compared to the single-frame cross-validation shown in Figure 3. It means that a larger value of $\lambda$ not necessarily have to lead to underfitting in as large extend.

When listening to the piano music, it is obvious that model improves the sound, reducing a lot of noise, but it is still far away from perfect. When trying a larger value of the capacity $\lambda = 0.02$, the noise cancelling is actually improved, the music is more clear. There is however another problem that the piano music have a different sound. Trying a smaller value of the hyperparameter $\lambda = 0.001$, the noise cancelling is performing worse with a lot of background noise again. I would say that the $\lambda$ selected with the K-fold cross-validation performs the best since it clearly reduces some background noise while still managing to keep the sound of the piano music more or less unchanged.

## References

[1] FMAN45 Machine Learning *Assignment Instructions: Assignment 1* Spring 2021. LTH. Downloaded: 2021-03-25. https://canvas.education.lu.se/courses/11000/files/1221084?wrap=1