Assignment 1

FMAN45 - Machine Learning

Author:
Dahlberg, Vilmer
vi8808da-s@student.lu.se

April 20, 2021

Introduction

In this report the LASSO method is studied, and is used to solve two regression problems. Firstly, a set of noisy data consisting of two main frequencies is to be denoised. The hyperparameter used in LASSO is first chosen using a guess-and-check method, but later trained using a cross-validation scheme. Next, a clip of music is denoised using LASSO, where the hyperparameter is trained using the same cross-validation scheme. The report consists of the seven tasks stated in the assignment, which are answered and discussed one by one.

Task 1

The LASSO problem can be solved using a coordinate-wise iterative method, where the i:th parameter is given by the solution to

$$\underset{w_i}{\text{minimize}}\ \frac{1}{2}||\boldsymbol{r_i} - \boldsymbol{x_i} w_i||_2^2 + \lambda |w_i|. \tag{1}$$

where $\boldsymbol{r}_i = \boldsymbol{t} - \sum_{l \neq i} \boldsymbol{x}_i w_l$ and $\boldsymbol{x}_i$ is the i:th row of the regression matrix $\boldsymbol{X}$.

Solution

Assume the j:th iterate of the i:th parameter is sought, meaning w_i^j. Let $\boldsymbol{r}^{(j-1)} = \boldsymbol{t} + \sum_{l<i} \boldsymbol{x}_l w_l^j + \sum_{l>i} \boldsymbol{x}_l w_l^{(j-1)}$ be the residual at the current step. Then, the next iterate is found by minimizing the function below with respect to w_i^j

$$f(w_i^j) = \frac{1}{2}||\boldsymbol{r}_i^{(j-1)} - \boldsymbol{x}_i w_i^j||_2^2 + \lambda |w_i^j| = \frac{1}{2}(\boldsymbol{r}_i^{(j-1)} - \boldsymbol{x}_i w_i^j)^T (\boldsymbol{r}_i^{(j-1)} - \boldsymbol{x}_i w_i^j) + \lambda |w_i^j|$$

As this is a convex problem, the minimum exists and is found by setting the derivative to zero. Assuming $w_i^j \neq 0 \iff \frac{d|w_i^j|}{dw_i^j} = \frac{w_i^j}{|w_i^j|}$, we get

$$\begin{aligned} \frac{df}{dw_i^j} &= -\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} + \boldsymbol{x}_i^T \boldsymbol{x}_i w_i^j + \lambda \frac{w_i^j}{|w_i^j|} = 0 \iff \\ w_i^j (\boldsymbol{x}_i^T \boldsymbol{x}_i &+ \lambda \frac{1}{|w_i^j|}) = \boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} \end{aligned} \tag{2}$$

To get an explicit expression for w_i^j, move the first term to the right-hand side and take the absolute value of the whole expression.

$$|w_i^j| \left| \boldsymbol{x}_i^T \boldsymbol{x}_i + \lambda \frac{1}{|w_i^j|} \right| = |\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| \tag{3}$$

Consider the left hand side of the expression. Using the fact that $\boldsymbol{x}_i^T \boldsymbol{x}_i, \lambda > 0$ we find

$$|w_i^j| \left| \boldsymbol{x}_i^T \boldsymbol{x}_i + \lambda \frac{1}{|w_i^j|} \right| = \left| \boldsymbol{x}_i^T \boldsymbol{x}_i |w_i^j| + \lambda \right| = \boldsymbol{x}_i^T \boldsymbol{x}_i |w_i^j| + \lambda$$

Finally solve for $|w_i^j|$ in equation 3, $|w_i^j| = \frac{1}{\boldsymbol{x}_i^T \boldsymbol{x}_i}(|\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| - \lambda)$, and insert into equation 2 to get

$$
\begin{aligned}
&w_i^j(\boldsymbol{x}_i^T \boldsymbol{x}_i + \lambda \frac{1}{|w_i^j|}) = \boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} && \iff \\
&w_i^j(\boldsymbol{x}_i^T \boldsymbol{x}_i |w_i^j| + \lambda) = |w_i^j| \boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} && \iff \\
&w_i^j |\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| = \frac{1}{\boldsymbol{x}_i^T \boldsymbol{x}_i}(|\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| - \lambda)\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} && \iff \\
&w_i^j = \frac{\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}}{\boldsymbol{x}_i^T \boldsymbol{x}_i |\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}|}(|\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| - \lambda)
\end{aligned}
$$

Task 2

Consider the case where the regression matrix is an orthonormal basis, i.e. $\boldsymbol{X}$ is square and $\boldsymbol{X}^T \boldsymbol{X} = I$. Show that the coordinate descent will converge in at most 1 iteration.

Solution

Begin by considering the update formula

$$
w_i^j = \begin{cases} \frac{\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}}{\boldsymbol{x}_i^T \boldsymbol{x}_i |\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}|}(|\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| - \lambda), & |\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)}| > \lambda \\ 0 & \text{else} \end{cases}
$$

where $\boldsymbol{r}^{(j-1)} = \boldsymbol{t} + \sum_{l<i} \boldsymbol{x}_l w_l^j + \sum_{l>i} \boldsymbol{x}_l w_l^{(j-1)}$. Using the fact that $\boldsymbol{X}$ is an orthonormal basis, meaning $\boldsymbol{x}_j^T \boldsymbol{x}_k = \delta_{jk}$, compute the product in the numerator.

$$
\boldsymbol{x}_i^T \boldsymbol{r}_i^{(j-1)} = \boldsymbol{x}_i^T \boldsymbol{t} + \sum_{l<i} \boldsymbol{x}_i^T \boldsymbol{x}_l w_l^j + \sum_{l>i} \boldsymbol{x}_i^T \boldsymbol{x}_l w_j^{(j-1)} = \boldsymbol{x}_i^T \boldsymbol{t}.
$$

Thus, the update formula becomes

$$
w_i^j = \begin{cases} \frac{\boldsymbol{x}_i^T \boldsymbol{t}}{|\boldsymbol{x}_i^T \boldsymbol{t}|}(|\boldsymbol{x}_i^T \boldsymbol{t}| - \lambda), & |\boldsymbol{x}_i^T \boldsymbol{t}| > \lambda \\ 0 & \text{else.} \end{cases}
$$

Which is obviously independent on the iteration number j, and hence the iteration converges in at most 1 iteration.

Task 3

Show that the LASSO estimate's bias, for an orthogonal regression matrix and data generated by $\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{e},\ \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}_N, \sigma \boldsymbol{I}_N)$, is given by

$$
\lim_{\sigma \to 0} E\left(\hat{w}_i^1 - w_i^*\right) = \begin{cases} -\lambda, & w_i^* > \lambda \\ -w_i^*, & |w_i^*| \leq \lambda \\ \lambda, & w_i^* < -\lambda \end{cases} \quad \forall i. \tag{4}
$$

Solution

As shown above when the regression matrix is orthogonal the update formula for the parameters w_i^j is given by

$$w_i^j = \begin{cases} \frac{\boldsymbol{x}_i^T \boldsymbol{t}}{|\boldsymbol{x}_i^T \boldsymbol{t}|}(|\boldsymbol{x}_i^T \boldsymbol{t}| - \lambda), & |\boldsymbol{x}_i^T \boldsymbol{t}| > \lambda \\ 0 & \text{else.} \end{cases}$$

Begin by considering the case where $|\boldsymbol{x}_i^T \boldsymbol{t}| \leq \lambda$, then

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

Next consider the case where $|\boldsymbol{x}_i^T \boldsymbol{t}| > \lambda$. The model used to generate $\boldsymbol{t}$ is known, so the product $\boldsymbol{x}_i^T \boldsymbol{t}$ can easily be computed

$$\boldsymbol{x}_i^T \boldsymbol{t} = \boldsymbol{x}_i^T (\boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{e}) = \sum_j \boldsymbol{x}_i^T \boldsymbol{x}_j w_j + \boldsymbol{x}_i \boldsymbol{e} = w_i^* + \boldsymbol{x}_i^T \boldsymbol{e},$$

where it was used that $\boldsymbol{x}_i^T \boldsymbol{x}_j = \delta_{ij}$. Now, study the limit of the bias as the variation in the noise tends to zero,

$$\lim_{\sigma \to 0} E\bigg(\hat{w}_i^1 - w_i^*\bigg) = \lim_{\sigma \to 0} E\bigg(\frac{w_i^* + \boldsymbol{x}_i^T \boldsymbol{e}}{|w_i^* + \boldsymbol{x}_i^T \boldsymbol{e}|}(|w_i^* + \boldsymbol{x}_i^T \boldsymbol{e}| - \lambda) - w_i^*\bigg).$$

Since the noise $\boldsymbol{e}$ has the expected value $\boldsymbol{0}$, the product $\boldsymbol{x}_i^T \boldsymbol{e}$ tends to zero as the variation in the noise tends to zero, implying $\boldsymbol{x}_i^T \boldsymbol{t}$ tends to w_i^*, thus

$$\lim_{\sigma \to 0} E\bigg(\hat{w}_i^1 - w_i^*\bigg) = E\bigg(\frac{w_i^*}{|w_i^*|}(|w_i^*| - \lambda) - w_i^*\bigg) = E\bigg(w_i^*(1 - \frac{\lambda}{|w_i^*|} - 1)\bigg) = E\bigg(-w_i^* \frac{\lambda}{|w_i^*|}\bigg) = \text{sign}(-w_i^*)\lambda.$$

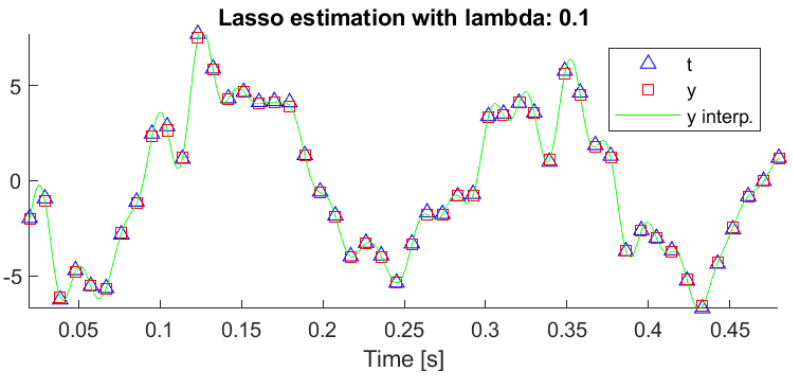
Divide the case $|\boldsymbol{x}_i^T \boldsymbol{t}| > \lambda$ into the two cases $\boldsymbol{x}_i^T \boldsymbol{t} > \lambda$, or $\boldsymbol{x}_i^T \boldsymbol{t} < -\lambda$, and use the fact that λ is positive.

$$\begin{cases} w_i^* > \lambda \implies w_i > 0 \implies & \lim_{\sigma \to 0} E\bigg(\hat{w}_i^1 - w_i^*\bigg) = \text{sign}(-w_i^*)\lambda = -\lambda \\ w_i^* < -\lambda \implies w_i < 0 \implies & \lim_{\sigma \to 0} E\bigg(\hat{w}_i^1 - w_i^*\bigg) = \text{sign}(-w_i^*)\lambda = \lambda \end{cases}$$

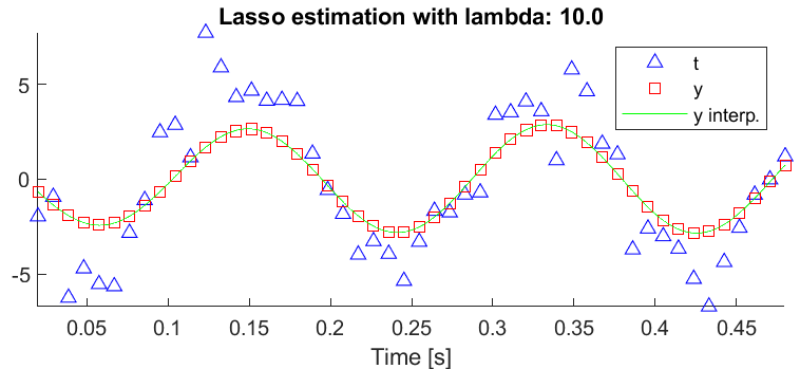
In words, the absolute value of the bias tends to λ if the absolute value of the weight is larger than λ, otherwise the estimation bias is the weight itself. This seems reasonable as the acronym LASSO stands for least absolute shrinkage and selection operator.

Task 4

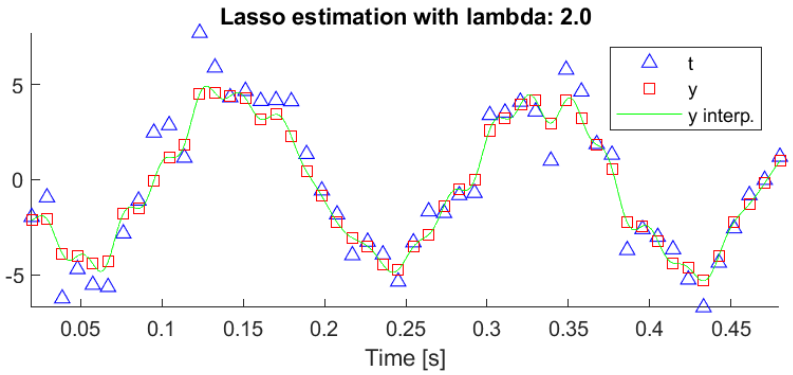
1. The LASSO estimation was implemented and tested using the given data. The results are shown below.



(a) Small penalty $lambda = 0.1$.



(b) A large penalty $lambda = 10$.



(c) A slightly smaller penalty $lambda = 2$.

Figure 1: LASSO estimation using different lambda values.

The penalty coefficient *lambda* determines how to weigh the error in the estimation and the magnitude of the coefficients w. If a small *lambda* is used, minimizing the error will be prioritized over minimizing the magnitude of the weights, and vice versa.

From the first plot, figure 1a, the reconstruction follows the data very closely. However, using this estimation for prediction would give poor results, since the data is overfit to include the random noise. A larger penalty should be used to decrease the amount of frequencies included.

In the next plot, figure 1b, the reconstruction seems to follow the low frequency oscillation well, but the high frequency oscillation is not followed. The reconstruction does not represent the model, and the error is too large. A smaller penalty should be used.

In the final plot, figure 1c, the reconstruction seems to follow both the low and high frequency oscillations well, although there is still some error. This lambda balances the minimization of the

error and the minimization of the magnitude of the coefficients well.

2. The number of nonzero coefficients were counted for each lambda value.

lambda	num. of $w \neq 0$
0.1	263
10	5
2	44

The number of coefficients needed to model the data is four, two for each frequency. However, around 44 coefficients are needed to get a good reconstruction of the data using this model. This also highlights the difficulty in choosing the penalty *lambda*.

Task 5

1. The figure below shows the root-mean-square error for the estimation folds in red and the validation folds in blue for different values of lambda. The optimal lambda is chosen as the lambda that minimizes the validation error.

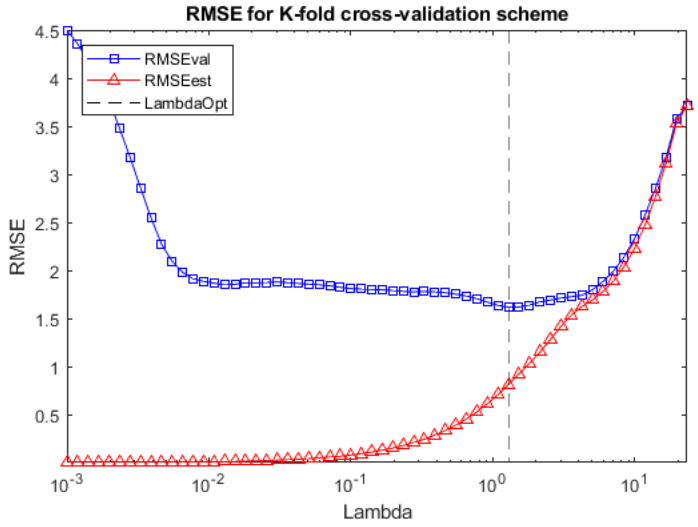


Figure 2: RMSE for validation and estimation folds. The lambda which minimizes $RMSE_{val}$ is chosen as the optimal lambda.

The estimation error grows as lambda increases, which is expected since an increase in lambda penalizes the magnitude of the coefficients and sacrifices the error. The validation error changes very little at and near the minimum, and choosing a lower lambda value would have a similar validation error but could reduce the estimation value by a significant amount.

Also note that the estimation error is always smaller than the validation error, which is especially clear when lambda is small. This is because a small lambda, hence a small penalisation, tends to result in an overfit of the data. If the data is overfit the estimation error shrinks, but the validation error grows since the noise is random. If the lambda is very small, the estimation error becomes very small, but the validation error grows. This is again due to the fact that the noise is random and an overfit of the noise in the estimation fold will result in large errors in the validation folds.

2. The plot shows a reconstruction of the data using the optimal lambda value. The number of nonzero w is still far too large. This suggests that the problem might lie in the model, and not the choice of lambda.

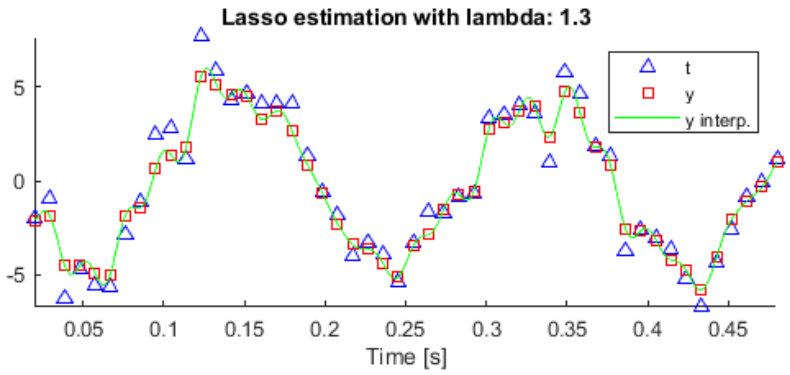


Figure 3: Number of nonzero w: 70.

Task 6

Just as in the previous task, the estimation error grows as the penalty increases, while the validation error has a clear minimum. Likewise, the estimation error is always lower than the validation error. Although the minimum is clearer in this case. The optimal lambda was about 4.9e-3.

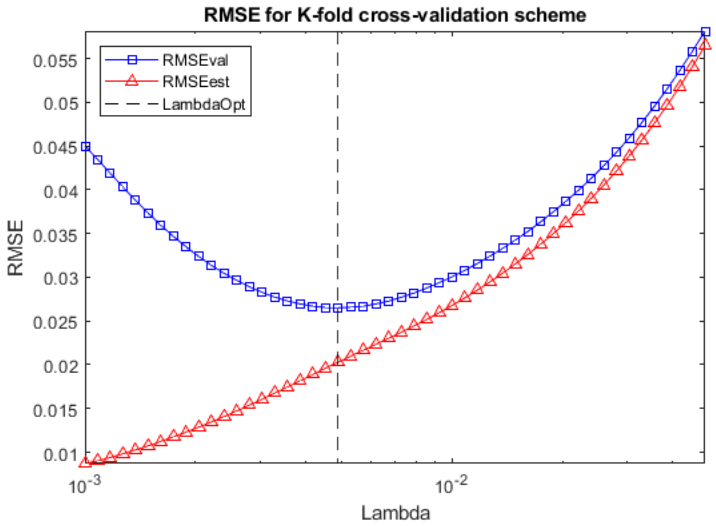


Figure 4: RMSE for validation and estimation folds. The lambda which minimizes $RMSE_{val}$ is chosen as the optimal lambda.

Task 7

A clear difference is heard when the cleaned audio is played after the noisy audio. However, the audio is still a bit noisy. Increasing the penalty reduces the noise, but reduces the amount of frequencies in the reconstruction which results in the piano sounding off. On the other hand, reducing the penalty makes the piano sound better but the noise comes back.

The method works, but has flaws and does not perform very well. A different approach is needed to remove all noise without destroying the piano sound. The method definitely works as a black-box method, but more information about the piano sound can be used. For example, the actual frequencies which comprise the piano sound are well known and can be used to further improve the quality in the cleaning of the audio.