FMAN45 Machine Learning - Assignment 1

Instructions: Read this assignment description carefully, solve the tasks stated, and typeset your solutions in e.g. LaTeX. Download the provided data and skeleton code, complete the programming tasks and submit your results and explanations in the same file as your theoretical answers. Present your solutions and results concisely in a detailed report. All necessary solutions, plots and figures should be in one self-contained pdf-document. It should thus be possible to understand all material presented in the report without running any code.

- Submit your pdf-document as well as the code you've written and data you've created via Canvas before the deadline.
- Typeset your solutions. Hand written solutions and scans/photos of hand written solutions will be rejected.
- Make sure all plots are readable even if they were printed on paper. Use differentiating colours, make sure that lines are thick enough, that markers are large enough to read without zooming in on your report and that legend are large enough.
- You should submit one pdf-document with your solutions and one zip-archive with your code. Do **not** include the code in the pdf-document unless explicitly stated.
- Do **not** include the pdf-document with the report in the archive.

To pass this assignment you must:

- Get at least 70 points in total.
- Get at least the number of points presented in brackets with each task.
- You can resubmit your assignment one time before summer if you get at least 30 points (total) on your first try. Exact deadline before the summer will be communicated through Canvas and the lectures.
- If you do not get 30 points on your first try or fail the resubmission you may resubmit in the re-exam period after the summer. Exact dates will be posted after the course has ended.

Also note that there are strict rules about collaboration and plagiarism, see information on assignment page.

Penalized regression via the LASSO

In linear regression, the purpose is to estimate the explanatory variable, or weights, $\mathbf{w} \in \mathbb{R}^M$ given a linear relationship to the response variable (data) $\mathbf{t} \in \mathbb{R}^N$, defined by the regression matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, i.e., the hypothesis space. In cases where for example the regression matrix is underdetermined and the model tends to overfit the data, or when a certain type of solution is required, a penalty term is added to the regression problem. In this assignment, you shall work with the so called Least Absolute Shrinkage and Selection Operator (LASSO), a type of penalised regression used when the sought explanatory variable is assumed to be sparse, i.e., having few non-zero coordinates. The LASSO solves the minimisation problem

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

where $\lambda \geq 0$ is a capacity hyperparameter. For the general regression matrix, there exists no closed-form solution for (1). Instead, consider a coordinate-wise approach, iteratively solving a sequence of optimisation problems, wherein each only one coordinate in $\mathbf{w}$ is optimised at a time, keeping the others fixed at their most recent value. Thus, for the i:th coordinate w_i, the objective is to

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

where $\mathbf{x}_i$ and $\mathbf{r}_i = \mathbf{t} - \sum_{\ell \neq i} \mathbf{x}_\ell w_\ell$ denotes the i:th vector of the regression matrix and the residual vector without the effect of the i:th regressor, respectively. Then, at iteration j, the coordinate descent minimizer of x_i has the closed-form solution

$$\hat{w}_i^{(j)} = \begin{cases} \frac{\mathbf{x}_i^\top \mathbf{r}_i^{(j-1)}}{\mathbf{x}_i^\top \mathbf{x}_i \left| \mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} \right|} \left(\left| \mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} \right| - \lambda \right), & \left| \mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} \right| > \lambda \\ 0, & \left| \mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} \right| \leq \lambda \end{cases} \tag{3}$$

$$\text{where } \mathbf{r}_i^{(j-1)} = \mathbf{t} - \sum_{\ell < i} \mathbf{x}_\ell \hat{w}_\ell^{(j)} - \sum_{\ell > i} \mathbf{x}_\ell \hat{w}_\ell^{(j-1)} \tag{4}$$

Exercise 1 - 20 (6) points: Verify the first line of Equation (3) by solving Equation (2) for $w_i \neq 0$. You do not need to show the condition, $\left|\mathbf{x}_i^\top \mathbf{r}_i^{(j-1)}\right| > \lambda$, for which it holds.
Hint: When solving for w_i after differentiating, you may want to get rid of w_i and solve for $|w_i|$ first. The absolute value $|w_i|$ is not differentiable for $w_i = 0$. However, for $w_i \neq 0$, its derivative is the sign of w_i, i.e.,

$$\frac{d\,|w_i|}{dw_i} = \frac{w_i}{|w_i|} \tag{5}$$

Now consider the simplified case where the regression matrix is an orthonormal basis, i.e., $M = N$, and $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_N$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix.

Exercise 2 - 10 (3) points: Given the orthogonal regression matrix, show that the coordinate descent solver in (3) will converge in at most 1 full pass over the coordinates in $\mathbf{w}$, i.e., show that $\hat{w}_i^{(2)} - \hat{w}_i^{(1)} = 0$, $\forall i$.
Hint: Starting from (3), you may use (4) to show that $\hat{w}_i^{(j)}$, for an orthogonal regression matrix, does not depend on previous estimates $\hat{\mathbf{w}}$, but only on $\mathbf{t}, \mathbf{x}_i$, and λ.

When $\mathbf{w}$ is estimated from data using the LASSO, a bias will be induced into it, i.e, $E\left(\hat{\mathbf{w}} - \mathbf{w}^*\right) \neq \mathbf{0}$. Assume that the data $\mathbf{t}$ is truly a noisy (stochastic) example from the hypothesis space defined by the regression matrix, that is

$$\mathbf{t} = \mathbf{X}\mathbf{w}^* + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}\left(\mathbf{0}_N, \sigma \mathbf{I}_N\right) \tag{6}$$

where $\mathbf{w}^*$ denotes the specific $\mathbf{w}$ defining the model used to generate the data, and where $\mathcal{N}(\cdot)$, $\mathbf{0}_N$, and σ denote the Gaussian distribution, a column vector of zeroes with length N, and a scalar denoting the standard deviation, respectively.

Exercise 3 - 10 (3) points: Show that the LASSO estimate's bias, for an orthogonal regression matrix and data generated by (6), is given by (7) when $\sigma \to 0$.

$$\lim_{\sigma \to 0} E\left(\hat{w}_i^{(1)} - w_i^*\right) = \begin{cases} -\lambda, & w_i^* > \lambda \\ -w_i^*, & |w_i^*| \leq \lambda \\ \lambda, & w_i^* < -\lambda \end{cases} \quad \forall i \tag{7}$$

Discuss how this result relates to the method's acronym, LASSO.
Hint: Starting from (3), you may consider the first line as two separate cases, $\mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} > \lambda$, and $\mathbf{x}_i^\top \mathbf{r}_i^{(j-1)} < -\lambda$, respectively.

Hyperparameter-learning via K-fold cross-validation

In this section, you shall consider LASSO regression for the noisy data `t` in `A1_data`, consisting of $N = 50$ data points generated by

$$t(n) = f(n) + \sigma \cdot e(n), \tag{8}$$
$$f(n) = \Re\left(5e^{i2\pi(\frac{n}{20}+\frac{1}{3})} + 2e^{i2\pi(\frac{n}{5}-\frac{1}{4})}\right), \tag{9}$$
$$e(n) \sim \mathcal{N}(0,1) \tag{10}$$

for $n = 0, \ldots, N-1$, i.e., the real part of a sum of two complex-valued sinusoids with frequencies $1/20$ and $1/5$ revolutions per sample, respectively, plus an additive Gaussian noise with standard deviation σ. The regression matrix you'll be using is given by `X` in `A1_data`, which consists of 500 candidate sine and cosine pairs, i.e.,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \ldots & \mathbf{X}_{500} \end{bmatrix} \tag{11}$$
$$\mathbf{X}_i = \begin{bmatrix} \mathbf{u}_i & \mathbf{v}_i \end{bmatrix} \tag{12}$$
$$\mathbf{u}_i = \begin{bmatrix} \sin(2\pi f_i 0) & \sin(2\pi f_i 1) & \ldots & \sin(2\pi f_i (N-1)) \end{bmatrix}^\top \tag{13}$$
$$\mathbf{v}_i = \begin{bmatrix} \cos(2\pi f_i 0) & \cos(2\pi f_i 1) & \ldots & \cos(2\pi f_i (N-1)) \end{bmatrix}^\top, \tag{14}$$

where f_i are the 500 frequency candidates, equally spaced on the interval $[0.02, 0.48]$.

Exercise 4 - 20 (6) points: Solve the three tasks below and discuss your results.

1. Implement a (cyclic) coordinate descent solver for the coordinate-wise LASSO solution in (3), using data `t` and regression matrix `X` in `A1_data`. Do **not** use `Xinterp` for estimation, it is provided for visualisation only.

   Hint: To simplify the implementation, you may fill in and use the function `skeleton_lasso_ccd()`.

2. Produce reconstruction plots with the following formatting: Plot the original $N = 50$ data points with disconnected markers (no connecting lines) vs. the time axis given by `n`. Overlay these with $N = 50$ reconstructed data points, i.e., $\mathbf{y} = \mathbf{X}\hat{\mathbf{w}}(\lambda)$, also using disconnected markers. In addition, add a solid line without markers for an interpolated reconstruction of the data, vs. an interpolated time axis. The interpolatation can be calculated using a highly sampled regression matrix, given as `Xinterp`. The interpolated time axis is found as `ninterp`. Specify legends for the all three lines, label the x-axis, and adjust line widths, font sizes and so on to make the plot clearly interpretable. Make three variants of this plot, one for each of the following values of the hyperparameter:

   - $\lambda = 0.1$.
   - $\lambda = 10$.
   - Try other values of λ to see the effect of the hyperparameter, and let $\lambda = \lambda_{\text{user}}$ be the one among these you find most suitable.

3. Count the number of non-zero coordinates for the values of the hyperparameter used above, i.e., $\lambda \in \{0.1, 10, \lambda_{\text{user}}\}$, and compare these to the actual number of nonzero coordinates needed to model the data given the true frequencies, which amounts to two pairs, or four coordinates.

   Hint: Look at Figure 1 at the end of this document and compare the signal with your reconstruction.

When $\lambda \to 0$, optimization problem Equation (1) reduces to an ordinary least squares problem, which if underdetermined, i.e., $M \geq N$, becomes ill-posed, and/or tends to overfit the data. The LASSO's penalty term promotes sparsity in $\mathbf{w}$ by setting coordinates with small magnitude to zero, thereby reducing the degrees of freedom for the problem. The choice of the hyperparameter λ thus becomes critical; if set too low then too many coordinates become non-zero, thus overfitting the data. If λ is set too high, then too many coordinates will be set to zero, thus underfitting the data. Let $\boldsymbol{\lambda}$ be a grid of J potential values of the hyperparameter, i.e.,

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_J \end{bmatrix} \tag{15}$$

where typically λ_1 is selected very small, and λ_J is selected as $\lambda_{\max}$, where,

$$\lambda_{\max} = \max_i \left(\left| \mathbf{x}_i^\top \mathbf{t} \right| \right). \tag{16}$$

To select λ optimally among these candidates, one may for instance use a K-fold cross-validation scheme. In this approach, the training data is randomly split in K non-overlapping and equally sized folds (subsets), where (K-1) folds are used for estimation, and the remaining fold is used for validation. More specifically, let the set $\mathcal{I}_k \subset \{1, \dots, N\}$ denote the $N_{\text{val}} = |\mathcal{I}_k|$ data indices selected for validation in the k:th fold. Thus, the complement of the validation index set, $\mathcal{I}_k^c \subset \{1, \dots, N\}$ denotes the set of $N_{\text{est}} = |\mathcal{I}_k^c| = N - N_{\text{val}}$ data indices used for estimation, such that, by definition, $\mathcal{I}_k \cap \mathcal{I}_k^c = 0$. This partitioning is repeated K times for each potential value of λ, cycling through the non-overlapping validation folds and using it for validation, while using the rest for estimation. To measure the performance of the estimated model when used to predict on the validation data, the root mean squared error on validation ($RMSE_{\text{est}}$) term

$$RMSE_{\text{val}}(\lambda_j) = \sqrt{K^{-1} \sum_{k=1}^{K} N_{\text{val}}^{-1} \left\| \mathbf{t}(\mathcal{I}_k) - \mathbf{X}(\mathcal{I}_k) \hat{\mathbf{w}}^{[k]}(\lambda_j) \right\|_2^2} \tag{17}$$

is used, where $\hat{\mathbf{w}}^{[k]}(\lambda_j)$ denote the weight estimate (herein the LASSO estimate) for the k:th fold using hyperparameter λ_j. Note that $\mathbf{X}(\mathcal{I}_k)$ refers to the (concatenation of) rows of $\mathbf{X}$ indexed by $\mathcal{I}_k$. The hyperparameter is then selected as

$$\hat{\lambda} = \lambda_p, \quad p = \operatorname*{argmin}_j \, RMSE_{\text{val}}(\lambda_j). \tag{18}$$

Algorithm 1 K-fold cross-validation for LASSO

Select grid of hyperparameters $\lambda_j \geq 0$, $\forall j$, and $K > 1$.
Intitialize $SE_{\text{val}}(k, \lambda_j) = 0$, $\forall k, j$.
Randomize validation indices $\mathcal{I}_k$, $\forall k$.
for k = 1, ..., K **do**
  for j = 1, ..., J **do**
    Calculate LASSO estimate $\hat{\mathbf{w}}(\lambda_j)^{[k]}$ on the k:th fold's estimation data.
    Set $SE_{\text{val}}(k, \lambda_j) = N_{\text{val}}^{-1} \left\| \mathbf{t}(\mathcal{I}_k) - \mathbf{X}(\mathcal{I}_k)\hat{\mathbf{w}}^{[k]}(\lambda_j) \right\|_2^2$.
  end for
end for
Calculate $RMSE_{\text{val}}(\lambda_j) = \sqrt{K^{-1} \sum_{k=1}^{K} SE_{\text{val}}(k, \lambda_j)}$.
Select $\hat{\lambda}$ as the λ_j which minimizes $RMSE_{\text{val}}$.

The following algorithm outlines a suggested implementation of a K-fold cross-validation scheme for the LASSO estimator:
To display the generalization gap, the corresponding root mean squared error on estimation data ($RMSE_{\text{est}}$) becomes

$$RMSE_{\text{est}}(\lambda_j) = \sqrt{K^{-1} \sum_{k=1}^{K} N_{\text{est}}^{-1} \left\| \mathbf{t}(\mathcal{I}_k^c) - \mathbf{X}(\mathcal{I}_k^c)\hat{\mathbf{w}}^{[k]}(\lambda_j) \right\|_2^2} \tag{19}$$

Exercise 5 - 20 (6) points: Implement a K-fold cross-validation scheme for the LASSO solver implemented in Task 4. Illustrate your results using the following plots and make detailed comments for them:

1. A plot illustrating $RMSE_{\text{val}}(\lambda_j)$ vs. $\lambda_j, \forall j$, using a solid line with markers. Overlaid in the same plot, also plot the corresponding $RMSE_{\text{est}}(\lambda_j)$ using a solid line with different markers. Also add a dashed vertical line at the location of the selected $\hat{\lambda}$. Specify legends for the all three lines, label the x-axis, and adjust line widths, font sizes and so on to make the plot clearly interpretable. Depending on your choice of λ-values you may want to use a logarithmic x-axis to make the plot more visible.

2. A reconstruction plot for the selected $\hat{\lambda}$. Use the same layout and formatting as described for the reconstruction plots in Task 4.

   Hint: For simpler implementation, you may fill in and use the function sketch `lasso_cv()`. Also, for improved vizualization and efficiency, select your grid of candidate λ to be evenly spaced on a log-scale. I MATLAB you can use `lambda_grid = exp( linspace( log(lambda_min), log(lambda_max), N_lambda))` while equivalent methods exist in Python.

Denoising of an audio excerpt

In general, the sounds of importance that we perceive in our daily lives are narrowband, meaning that for any short (< 100 ms) sequence, the spectral representation is sparse, i.e., it may be modeled using only a small number of pure frequencies. This is typically true for, e.g., speech and music. As a contrast, sounds of small importance, or even nuisance sounds, are often broadband, such as, e.g., noise and interference, containing many or most frequencies across the spectrum.

In this section, you shall attempt to perform denoising of a noisy recording of piano music using the LASSO and a regression matrix of sine and cosine pairs. The idea is thus that, using a hypothesis space with frequency functions, and by virtue of the sparse estimates obtained using the LASSO, only the strong narrowband components in the music will be modeled, while the broadband noise is cancelled.

The data archive `A1_data` contains an excerpt of piano music, 5 seconds

in total, divided into two data sets, `Ttrain`, and `Ttest`, where the former is for estimation and validation, while the latter is used only for testing (**not** validation). The training data is much too large (and non-stationary) to be modeled in one regression problem. Instead, the training data is to be divided into frames 40 ms long. For each frame, a (possibly unique) LASSO solution may be obtained, which if used to reconstruct the data can be listened to. The same regression matrix, `Xaudio` is used for all frames. Do **not** use regression matrix `X` from the previous section.

In order to listen to the data you can use:

- `soundsc(Ttrain,fs)` in MATLAB, where `fs` is the sampling rate, also provided in the data archive.
- `sounddevice.play(Ttrain, fs)` in Python, `fs` is the sampling rate, where sounddevice can be installed via e.g. pip. There are also other libraries for this use

Exercise 6 - 10 (3) points: Implement a K-fold cross-validation scheme for the multi-frame audio excerpt `Ttrain`, similar to Task 5, but modified in order to find one optimal $\hat{\lambda}$ for all frames. Illustrate the results with a plot of $RMSE_{\text{val}}$, $RMSE_{\text{est}}$, and $\hat{\lambda}$, using the exact same formatting as in the corresponding plot in Task 5, and discuss your findings.
Hint: The provided function sketch `multiframe_lasso_cv()` outlines the algorithm, which may be filled in and used. For the data provided, complete training may take 15 minutes or longer, depending on the implementation.

Exercise 7 - 10 (3) points: Using the $\hat{\lambda}$ you've trained in Task 6, use the provided function `lasso_denoise()` to denoise the test data `Ttest`. Save your results as `save('denoised_audio','Ytest','fs')` in MATLAB or as an ordinary NumPy-array if you are using sounddevice in Python and include it in your submission. In addition, listen to your output discuss your results. Try whether another choice of λ achieves better noise-cancelling results and detail your findings in the report.

Appendix

A Provided data and functions

Contents of data archive `A1_data`:

`t`	Single realization of data $\mathbf{t} \in \mathbb{R}^{50}$ in Equation (8)
`X`	Regression matrix for Task 4 and 5, $\mathbf{X} \in \mathbb{R}^{50\times1000}$ in Equation (11)
`Xinterp`	Like `X` but with interpolated time axis $\mathbf{X} \in \mathbb{R}^{1000\times1000}$
`n`	Time axis $\mathbf{n} \in \mathbb{Z}^{50} : n \in [0, N-1]$
`ninterp`	Interpolated time axis $\mathbf{n} \in \mathbb{R}^{1000} : n \in [0, N-1]$
`Ttrain`	Mono audio data for estimation and validation, $\mathbf{T} \in \mathbb{R}^{19404}$
`Ttest`	Mono audio data for testing, $\mathbf{T} \in \mathbb{R}^{24697}$
`Xaudio`	Regression matrix for Tasks 6 and 7, $\mathbf{X} \in \mathbb{R}^{364\times2000}$ in Equation (11)
`fs`	Sampling frequency, $f_s = 8820$ Hz.

Function sketches (partly contains pseudocode) provided:

- `what = lasso_ccd(t,X,lambda)`:

  Returns LASSO estimate `what` (i.e., $\hat{\mathbf{w}}(\lambda)$) given an input of data `t`, regression matrix `X`, and hyperparameter `lambda`.

- `[wopt,lambdaopt,RMSEval,RMSEest] = lasso_cv(t,X,lambdavec,K)`:

  Returns `wopt`; the LASSO estimate for the optimal $\hat{\lambda}$, as well as `lambdahat`; the optimal lambda, and `RMSEval`, `RMSEest` the $RMSE$ on estimation and validation, respectively. Takes `t`, regression matrix `X`, grid of candidate hyperparameters `lambdavec`, and the number of folds `K`, as inputs.

- [Wopt, lambdahat, RMSEval, RMSEest] = multiframe_lasso_cv(T,X,lambdavec,K)

  Returns `Wopt`; an `M` by `N_frames` matrix of LASSO estimates for each size-`N` frame of data in `T`, calculated for the optimal $\hat{\lambda}$, as well as `lambdahat`; the optimal lambda, and `RMSEval`, `RMSEest` the $RMSE$ on estimation and validation, respectively. Takes multi-frame data `T`, regression matrix `X`, grid of candidate hyperparameters `lambdavec`, and the number of folds `K`, as inputs.

- `Yclean = lasso_denoise(T,X,lambdaopt)`:

  Returns the denoised audio vector `Yclean` given an input of noisy audio `T`, regression matrix `X`, and hyperparameter `lambdaopt`.

B Illustration of data in Task 4 and 5

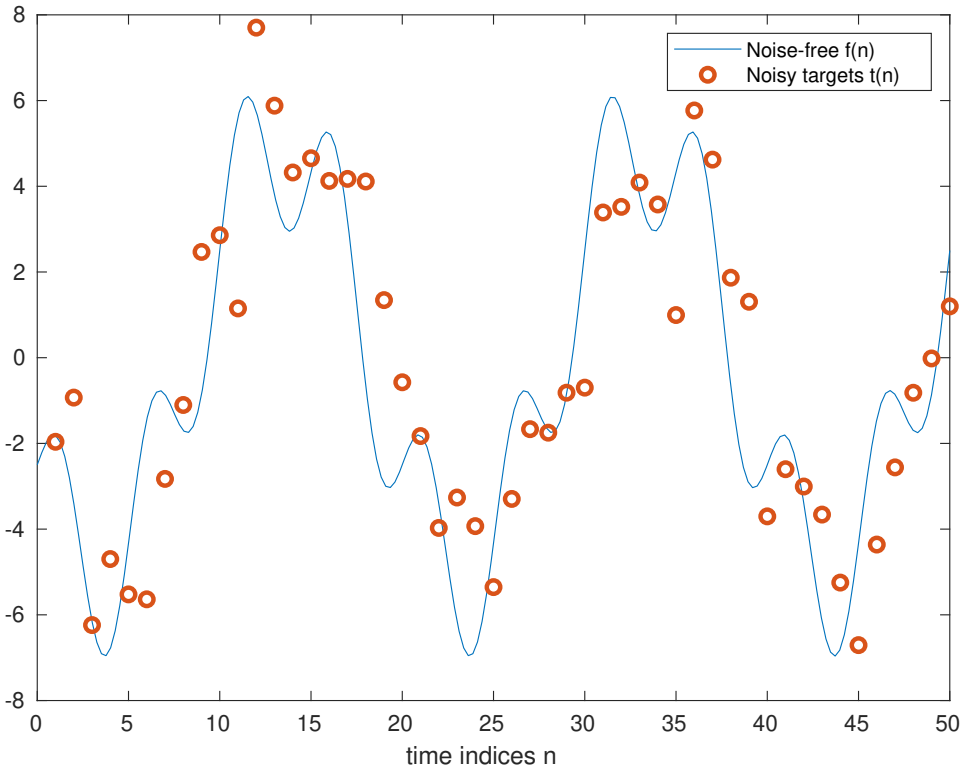

Figure 1: Plot of the provided noisy target values (data) `t`, together with the noise-free data generating function $f(n)$ in (9). A good reconstruction should look somewhat close to this!