# Machine Learning

*Assignment 1*

**Submitted by:**

**Vahid Faraji**

Master's Student in Machine Learning, Systems and Control
Lund University, Sweden

**Submission Date:** April 11, 2025

# 3 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.

> **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.
> **Hint 2:** You may, for example, centre your discussion around how the errors propagate with different $\lambda$. Is the behaviour expected?

### 3.0.1 Solution

The $\lambda$ that gave the lowest validation RMSE was $\lambda = 0.0049$, so this was selected as the optimal value (denoted $\hat{\lambda}$). In Figure 3, we can see that the validation RMSE reaches its minimum at this $\lambda$ (the plot marks the chosen $\hat{\lambda}$).

For very small values of $\lambda$ (almost no regularization), the model tends to overfit the training data. This causes a high RMSE on the validation set because the model does not generalize well. As $\lambda$ increases to a moderate range, the model is more constrained (less overfitting), and the validation RMSE decreases. It reaches the lowest point at an intermediate $\lambda$ (around 0.0049 in this case). If $\lambda$ becomes very large, the model is over-regularized and cannot fit the data well (underfitting), which makes the RMSE go up again.

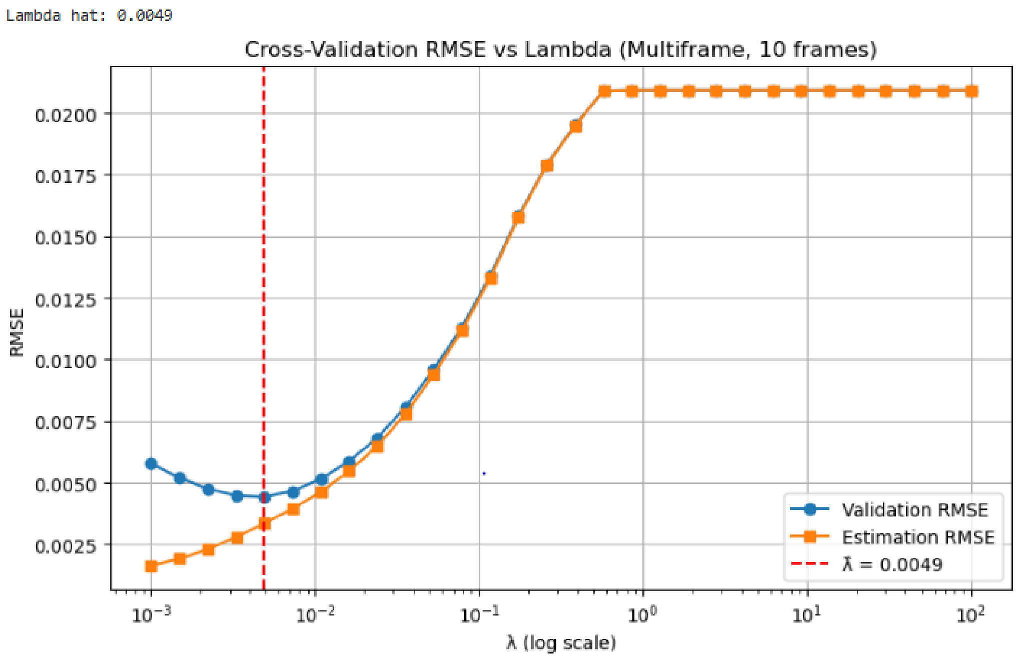


Figure 3: Validation RMSE and Estimation RMSE vs. $\lambda$. The red dash line:$\hat{\lambda} = 0.0049$

---

> **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 $\lambda$ achieves better noise-cancelling results and detail your findings in the report. **Hint:** You may, for example, centre your discussion regarding the results for different values of $\lambda$, the objectiveness of the found $\lambda$ or some other topic.

### 3.0.2 Solution

In this task, I used the $\hat{\lambda} = 0.0049$ that was selected in Task 6 to denoise the test signal $T_{test}$ using the `lasso_denoise()` function. The output signal was saved as `Ytest.npy`.

I also tried some other values of $\lambda$ (like 0.001, 0.01, 0.05, and 0.1) and listened to the results. When $\lambda$ was too small (like 0.001), there was still a lot of noise. When $\lambda$ was too big (like 0.05 and especially 0.1), the sound was relatively changed and some of the piano tone was missed.

Regarding audiotory results, the best results were for $\lambda = 0.0049$ and $\lambda = 0.01$. It can be visually seen from Figure 3 that RMSE does not experience very large variation in this region. So, it is justifiable that both of them had low noise and good audio quality. This shows that the $\hat{\lambda}$ from Task 6 also works well in practice, not just in RMSE.