Mini-Project (ML for Time Series) - MVA 2024/2025

Antoine Martinez antoine.martinez.x21@polytechnique.edu
Mathias Grau mathias.grau.@polytechnique.edu

December 17, 2024

1 Introduction and contributions

This article [1] explores dictionary learning for signal reconstruction using training data. The authors first show the similarity between dictionary-based signal reconstruction and clustering, where a clustering problem can be seen as an extreme case of dictionary learning, assigning a signal to a single atom with a coefficient of 1. The K-means method, a standard approach to clustering, adapts cluster barycenters as it solves the problem, which is of interest for dictionary adaptation. Rather than using a fixed dictionary, the goal is to adapt its atoms to better suit the specific dataset, generalizing the K-means method for this purpose.

2 Method

The K-SVD method aims to solve the following optimization problem:

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \quad \text{s.t.} \quad \forall i, \quad \|\mathbf{x}_i\|_0 \leq T_0 \tag{1}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{n \times N}$ represents N signals to be approximated, $\mathbf{D} \in \mathbb{R}^{n \times K}$ is the dictionary containing K atoms, and $\mathbf{X} \in \mathbb{R}^{K \times N}$ contains the coefficients relating atoms to signals. The Frobenius norm $\|.\|_F$ measures the reconstruction error, and $\|.\|_0$ denotes the number of non-zero values in the vector.

The method consists of two main steps: a sparse coding stage and a dictionary update stage using Singular Value Decomposition (SVD). The latter is applied to each of the K atoms in the dictionary, hence the name K-SVD.

2.1 Sparse coding step

In this stage, the dictionary $\mathbf{D}$ is fixed, and the goal is to approximate each of the N signals in $\mathbf{Y}$ using the best T_0 out of the K atoms. This is typically achieved through methods such as Matching Pursuit, Orthogonal Matching Pursuit [2, 3], Basis Pursuit, or the Lasso.

The problem (1) can be decoupled into N distinct sub-problems:

$$\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq T_0 \quad \forall i \in [N] \tag{2}$$

2.2 Dictionary update

The main contribution of this article is the sequential update of each dictionary atom using Singular Value Decomposition (SVD). Focusing on a specific atom d_k, the error matrix can be rewritten as follows, using $\mathbf{DX} = \sum_{k=1}^{K} d_k \mathbf{x}_T^k$ (Appendix 6.1.1):

$$\|\mathbf{Y} - \mathbf{DX}\|_F = \|\mathbf{Y} - \sum_{j=1}^{K} d_j \mathbf{x}_T^j\|_F = \|\mathbf{Y} - \sum_{j=1, j\neq k}^{K} d_j \mathbf{x}_T^j - d_k \mathbf{x}_T^k\|_F = \|\mathbf{E}_k - d_k \mathbf{x}_T^k\|_F \tag{3}$$

The SVD decomposition of $\mathbf{E}_k$ is used to update d_k and $\mathbf{x}_T^k$ sequentially for all K atoms in $\mathbf{D}$. To maintain sparsity constraints, a variable ω_k is introduced to identify signals $\mathbf{y}_i$ that use atom d_k:

$$\omega_k = \{i \in [N] \mid \mathbf{x}_T^k(i) \neq 0\} \tag{4}$$

A matrix $\Omega_k \in \mathbb{R}^{N \times |\omega_k|}$ is computed, where $[\Omega_k]_{\omega_k(i),i} = 1$ for all $i \in \omega_k$ and 0 elsewhere (Figure 1). This results in reduced versions of the error matrix and activation vector for atom k:

$$\mathbf{X}_R^k = \mathbf{X}_T^k \Omega_k \qquad \mathbf{Y}_k^R = \mathbf{Y}\Omega_k \qquad \mathbf{E}_k^R = \mathbf{E}_k \Omega_k$$

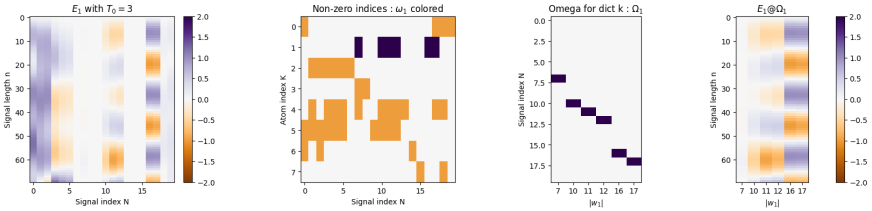


Figure 1: Creation of Ω_k ($k = 1$) and the reduced error matrix

This step optimizes the dictionary and activation values while respecting sparsity constraints. The minimization of Equation (3) with sparsity constraints becomes:

$$\|\mathbf{E}_k \Omega_k - d_k \mathbf{x}_T^k \Omega_k\|_F^2 = \|\mathbf{E}_k^R - d_k \mathbf{x}_R^k\|_F^2 \tag{5}$$

This operation reduces the matrices by removing coefficients not used in the update of d_k and $\mathbf{x}_T^k$ due to the sparsity of $\mathbf{x}_T^k$. For example, $\mathbf{X}_R^k$ retains only the non-zero coefficients of $\mathbf{x}_T^k$. The SVD decomposition on $\mathbf{E}_k^R$ is then computed to optimize (5):

$$\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T \tag{6}$$

We update d_k as the first column of $\mathbf{U}$ and $\mathbf{x}_R^k = \Delta_{1,1}\mathbf{V}_1$. This operation preserves the normalization of d_k. Finally, $\mathbf{x}_T^k$ is recovered through the transpose of Ω_k. The key innovation is the simultaneous update of both the dictionary atom d_k and the activation vector $\mathbf{x}_T^k$.

Complexity: The complexity of the method arises from the sparsity update and the K-SVD update. For the sparsity update, using Orthogonal Matching Pursuit (OMP), the complexity is $\mathcal{O}(T_0(KN + KT_0^2 + T_0^3))$. For the K-SVD update, a Singular Value Decomposition (SVD) is performed for each dictionary atom d_k on the reduced error matrix $\mathbf{E}_k^R$. In the worst-case scenario, where the atom is active for all signals, the complexity of the SVD is $\mathcal{O}(n^2N)$. Consequently, the overall complexity of the K-SVD update is $\mathcal{O}(Kn^2N)$.

Convergence considerations: The algorithm guarantees monotonic reduction in representation error under ideal conditions, ensuring convergence to a local minimum. The success of this process relies on the performance of pursuit algorithms like OMP, FOCUSS, and BP, which are effective when the sparsity level is sufficiently small relative to the signal dimensions. Experiments demonstrated that convergence is naturally achieved without additional measures, confirming the monotonic reduction in error at each stage.

3 Data

3.1 Synthetic experiments

We reproduced one experiment of the article where the goal was to measure the capacity of a method to find out the dictionary which produced all the signals of the input. To that extend, we generated $K = 50$ signals of length $n = 20$ with a normal distribution,then normalized each signal,and we get our "true" dictionary $\mathbf{D}$ with which we would compare the learned dictionary of the methods. Then, for each of the $N = 1500$ signals of $\mathbf{Y}$, we selected uniformly T_0 atoms from $\mathbf{D}$, (T_0 different signals). For each of the chosen atoms we sample from a uniform distribution a scalar $\lambda \in [0, 3]$ as a weight, and we got $\mathbf{Y}$ composed of N signals each of them produced with T_0 atoms of $\mathbf{D}$. We also added some noise to the input signals to see its impact. The goal of this experiment was to count how many atoms the methods had recognized in its final dictionary fitted on the $\mathbf{Y}$ signals. To that extend, we compared the learned dictionary of each method with the "true" dictionary. This comparison was done by sweeping through the columns of the generating dictionary and finding the closest column, measuring the distance via:

$$\forall i, j \in [K], \quad Dist(d_i, \hat{d}_j) = 1 - |d_i^T \hat{d}_j|$$

And we considered that it was a match : ie the method recognized an atom if this distance was lower than 0.01.

In order to compare this method with another Dictionary Learning method, we implemented the Method of Optimal Direction (MOD), which is also composed of two steps. A sparse coding step (also done with OMP), in order to update the activation vector $\mathbf{X}^{(n)}$. Followed by a dictionary update at each step $n \in \mathbb{N}$: $\mathbf{D}^{(n+1)} = \mathbf{Y}\mathbf{X}^{(\mathbf{n})^T}(\mathbf{X}^{(\mathbf{n})}\mathbf{X}^{(\mathbf{n})^T})^{-1}$.

3.2 Image processing

Afterwards, we implemented a full pipeline for image processing and denoising. We started from impressionist paintings of size 256×256 pixels presented in Figure 2(a). We created patches of size 8×8 to split images in different regions. A plot of these patches can be found in Figure 2(b) where these patches are ranked by increasing variance.

(a) Impressionists paintings

(b) 500 random train patches sorted by increasing variance

Figure 2: Training paintings (256×256) and 500 extracted patches (8×8) for the first channel

4 Results

4.1 Synthetic results

We tested the experiment described above to measure the effectiveness of the method in recognizing the atoms in the dictionary, with a maximum number of iterations fixed at 80. The results are presented in Figure 3 below:

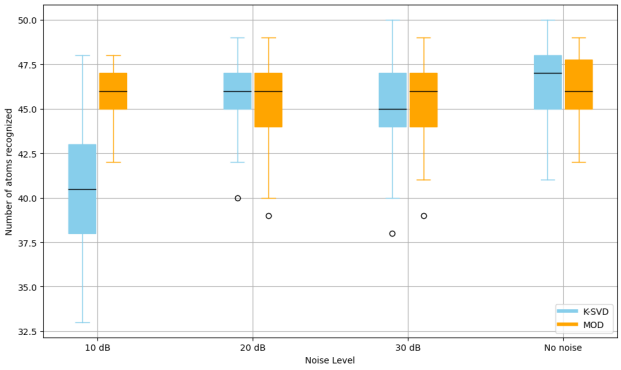


Figure 3: Number of atoms detected by K-SVD and MOD

If we look at the lowest noise levels, we can see that both methods have the same results, although K-SVD performs slightly better when noise is very low or non-existent. On the other hand, when the noise level rises, in this case to 10 dB, the MOD method seems more resilient, and its performance drops less than that of K-SVD. Surprisingly, this pattern was not in the article's plot, perhaps due to differences in the experiment's parameter settings. Nevertheless, we can say that the results of the experiment seem to be very satisfactory for our K-SVD method, as it does a very good job of recognizing the atoms in the basic dictionary.

This was done to assess the impact of the key parameters of this method in this experiment. We therefore varied the size of the signals, the maximum number of iterations allowed, and the number of signals N in $\mathbf{Y}$. We plotted the results of these simulations in the figures in the Appendix, and we can see that the maximum number of iterations does not really influence performance as long as it is not too low 5(a), so we chose to stay with it equal to 80 for optimum accuracy while keeping runtime low. As far as the signal length is concerned, an increase in signal length results in much better performance 5(b), with a very strong improvement between lengths 13 and 17, where the increase is linear with a very steep slope. Finally, as we would expect, the number of signals N in $\mathbf{Y}$ also has a positive impact on the recognition 5(c). Therefore, at a certain point, increasing this number is useless, which is why we left it at its first value of 1500.

4.2 Completion of missing pixels

The $N = 11000$ patches from Figure 2(b) were flattened to create signals of length $n = 64$. We then trained our K-SVD model to construct a dictionary of size $K = 441$ based on these signals. The resulting trained patches extracted from the dictionary are displayed in Figure 6 (Appendix).

Subsequently, we evaluated our dictionary's performance in reconstructing images with missing pixels. We corrupted our test images (*Still Life with Pears and Grapes* by Claude Monet) by removing a large proportion of pixels (setting them to 0), then created patches and used our dictionary to recover the original image. During the sparse coding step, only the non-zero values of the signals and the corresponding dictionary values were used to find the sparse activations. Practically, while the original signals had a length of $n = 64$, we fit the dictionary only to the non-zero pixel values, effectively reducing the signal length to $n' \leq n$. Sparse activations were computed based on these reduced signals, and the full signal was then recovered. The results of this reconstruction process are presented in Figure 4.

(a) 50% of missing pixels : RMSE = 0.0042

(b) 70% of missing pixels : RMSE = 0.0081

Figure 4: Reconstruction of an image with varying proportion of missing pixels (50% and 70%)

5 Conclusion

This study introduced the K-SVD method for dictionary learning in signal reconstruction, inspired by the K-means clustering algorithm. The method involved sparse coding to approximate the best dictionary atoms for each signal and a dictionary update using Singular Value Decomposition (SVD) to adaptively refine the atoms while maintaining sparsity constraints. Experimental results validated K-SVD's effectiveness, showing strong performance in recognizing dictionary atoms, particularly in low-noise environments, and robustness in image processing tasks, successfully reconstructing images with a high proportion of missing pixels.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *IEEE*, pages 40–44 vol.1, 1993.

[3] S. A. BILLINGS S. CHEN and W. LUO. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989.

6 Appendix

6.1 Proofs

6.1.1 Selection of specific atom

If we fix $\mathbf{X}$ and $\mathbf{D}$, except for the d_j column of $\mathbf{D}$, which represents the j^{th} atom of the dictionary, we can rewrite the problem to explicitly show d_j. For this, we denote $\mathbf{x}_T^j$ as the j^{th} row of $\mathbf{X}$ and observe that:

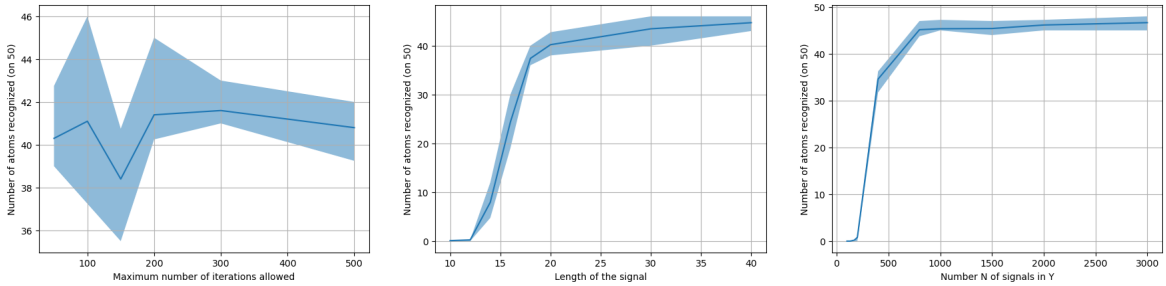
$$\forall i \in [n], \forall j \in [N], \quad \left[\sum_{k=1}^{K} d_k \mathbf{x}_T^k\right]_{i,j} = \sum_{k=1}^{K} [d_k \mathbf{x}_T^k]_{i,j} = \sum_{k=1}^{K} d_{i,k} \mathbf{x}_{k,j} = [\mathbf{DX}]_{i,j}$$

Thus, we have $\mathbf{DX} = \sum_{k=1}^{K} d_k \mathbf{x}_T^k$. This leads to the equation introduced in (3):

$$\|\mathbf{Y} - \mathbf{DX}\|_F = \|\mathbf{Y} - \sum_{j=1}^{K} d_j \mathbf{x}_T^j\|_F = \|\mathbf{Y} - \sum_{j=1, j\neq k}^{K} d_j \mathbf{x}_T^j - d_k \mathbf{x}_T^k\|_F = \|\mathbf{E}_k - d_k \mathbf{x}_T^k\|_F \tag{7}$$

6.2 Influence of the parameters

All experiments to assess the impact of different parameters were conducted using the following settings: $K = 50$, $N = 1500$, $n_{\text{signal}} = 20$, $\text{SNR}_{\text{noise}} = 30$ dB, $\max_{\text{iter}} = 80$, $T_0 = 3$, except for the specific parameter being varied in each experiment.



(a) Variation of recognition with maximum iterations allowed

(b) Variation of recognition with signal length

(c) Variation of recognition with the number of signals in $\mathbf{Y}$

Figure 5: Performance tests on recognition

6.3 Learned patches from impressionist dataset

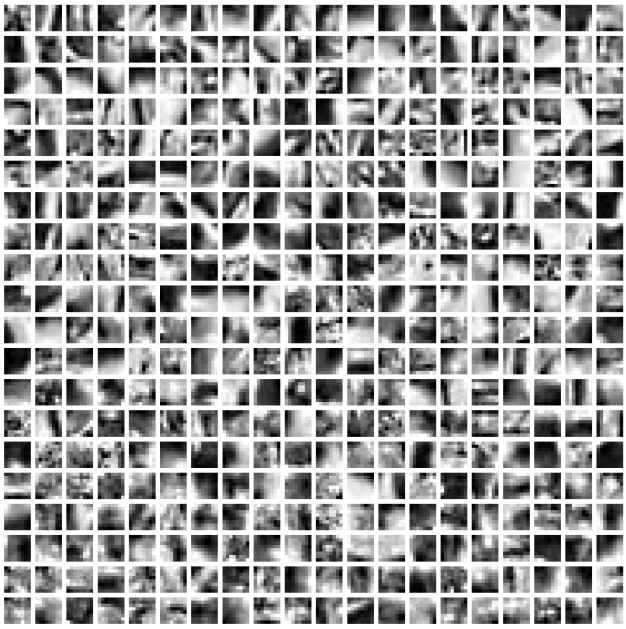
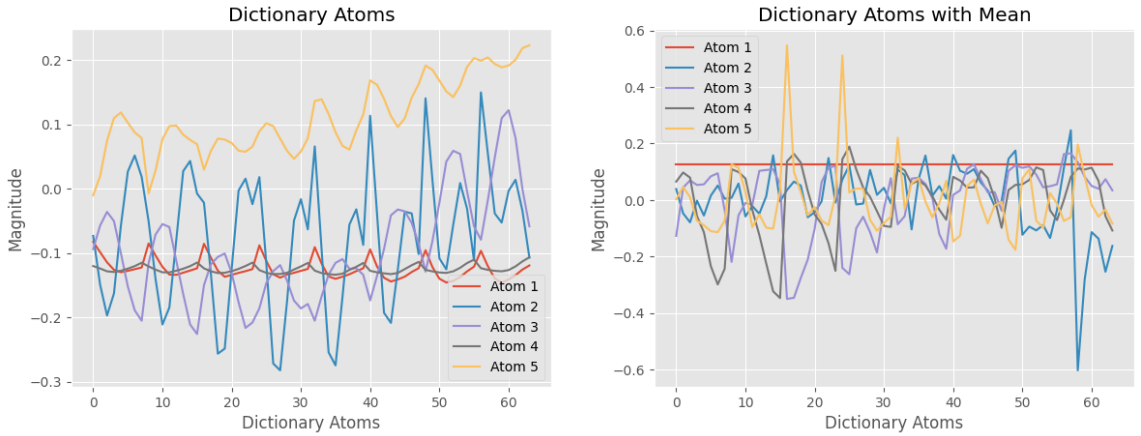


Figure 6: Learned patches sorted by increasing variance

6.4 K-SVD dictionary learning method comparison



(a) Simple dictionary atoms

(b) Dictionary atoms (first constant, others centered)

Figure 7: First atoms of dictionaries

The second method enhances reconstruction by keeping the first atom constant. When combined with the corresponding activation, it encodes the mean of each signal, ensuring that the other atoms are centered.

(a) Original Image (b) Reconstructed Image (without centering) (c) Reconstructed Image (with centering)

Figure 8: Example reconstruction of an image using $K = 20$ atoms and 1) a simple dictionary 2) a dictionary learned from normalized signals with one constant atom

Using dictionary atoms with the first set to a constant value enhances the quality of the reconstructed image. This approach ensures that the mean of each signal is encoded, leading to better centering of the other atoms and improved overall reconstruction.