K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation

Mathias Grau, Antoine Martinez

December 18th, 2024

Introduction and contributions

K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation (2006) by Michal Aharon, Michael Elad, and Alfred Bruckstein. [1]

- Dictionary learning based on training signal data
- Clustering problem (K-means method)
- Successive optimization alternating between sparse coding and dictionary update

Outline

1. Method
   - Sparse coding step
   - Dictionary update
   - Complexity and convergence
2. Data
   - Synthetic experiments
   - Image processing
3. Results
   - Synthetic results
   - Completion of missing pixels

Problem

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

With :

- N signals which we want to approach
- n the length of each signals and dictionary atoms
- K the number of atoms in the dictionary
- T_0 the sparsity, number of atoms allowed to reconstruct each signals
- $\mathbf{Y} = [\mathbf{y_1}, \ldots, \mathbf{y_N}] \in \mathbb{R}^{n \times N} \quad \forall i, \quad \mathbf{y_i} \in \mathbb{R}^n$
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ the dictionary of K atoms'
- $\mathbf{X} \in \mathbb{R}^{K \times N}$, the coefficients between atoms and signals

Step 1 : Sparse coding step

- **D** is fixed
- Approximate the best T_0 among the K atoms
- Matching Pursuit, Orthogonal Matching Pursuit [3] [2], Basis Pursuit, or the Lasso, among others.
- K-SVD let the user select the method

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

Step 2 : Dictionary update

- Idea : Update one atom of the dictionary at a time using SVD decomposition
- Error matrix with a focus on a specific atom d_k

$$\|\mathbf{Y}-\mathbf{DX}\| = \|\mathbf{Y}-\sum_{j=1}^{K} d_j \mathbf{x}_T^j\|_F = \|\mathbf{Y}-\sum_{j=1, j\neq k}^{K} d_j \mathbf{x}_T^j - d_k \mathbf{x}_\mathbf{T}^\mathbf{k}\|_F = \|\mathbf{E_k} - d_k \mathbf{x}_T^k\|_F \quad (3)$$

- Idea : use the SVD decomposition of $\mathbf{E_k}$ to update d_k and $\mathbf{x}_T^k$, for $k = 1, \ldots, K$

Step 2 : Dictionary update (Idea)

- Warning : sparsity constraints of the activation vectors $\mathbf{x}_T^k$

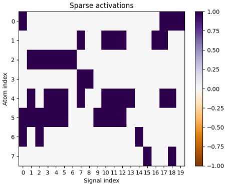


Figure: Sparse activations (matrix $\mathbf{X}$)

- New variable ω_k that checks for all signals $\mathbf{y}_i$ if they use the atoms d_k : $\omega_k = \{i \in [N] \quad | \quad \mathbf{x}_T^k(i) \neq 0\}$
- $\Omega_k \in \mathbb{R}^{N \times |\omega_k|}, \quad \forall i \in [|\omega_k|] \quad [\Omega_k]_{\omega_k(i),i} = 1$ and 0 everywhere else

Step 2 : Dictionary update (Sparsity selection)

- We have access to reduced versions to be updated :

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

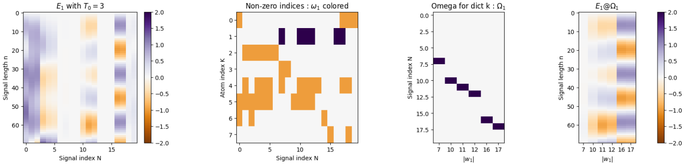


Figure: Creation of Ω_k ($k = 1$) and the reduced error matrix

- Equation (3) minimization with sparsity constraints becomes

$$\|\mathbf{E}_k \Omega_k - d_k \mathbf{x}_T^k \Omega_k\|_F^2 = \|\mathbf{E}_k^R - d_k \mathbf{x}_R^k\|_F^2 \tag{4}$$

Step 2 : Dictionary update (SVD)

$$\|\mathbf{E}_k \Omega_k - d_k \mathbf{x}_T^k \Omega_k\|_F^2 = \|\mathbf{E}_k^R - d_k \mathbf{x}_R^k\|_F^2 \tag{4}$$

- Compute the SVD decomposition of $\mathbf{E}_k^R$

$$\mathbf{E}_k^R = \mathbf{U} \boldsymbol{\Delta} \mathbf{V}^T \tag{5}$$

- Update on d_k and $\mathbf{x}_R^k$ as follow :

$$d_k = \mathbf{U}_1 \qquad \mathbf{x}_R^k = \Delta_{1,1} \mathbf{V}_1$$

- The operation preserve normalization on d_k.
- We update the dictionary d_k **AND** the activations $\mathbf{x}_T^k$.

Complexity and convergence

Complexity :

- Sparsity update :
  - Depends on the method selected
  - Orthogonal Matching Pursuit : $\mathcal{O}(T_0(KN + KT_0^2 + T_0^3))$
- K-SVD update
  - Singular Value Decomposition
  - Need of singular vectors : perform all the matrices U and V
  - $\mathcal{O}(Kn^2N)$ (if $n \leq N$)

Convergence :

- Ideal conditions, such as perfect sparse coding
- Ensure **convergence to a local minimum**
- Convergence relies heavily on the performance of pursuit algorithms like OMP, FOCUSS, and BP
- Experimentally, we confirmed the **monotonic reduction** in error at each stage 9

Outline

Synthetic experiments

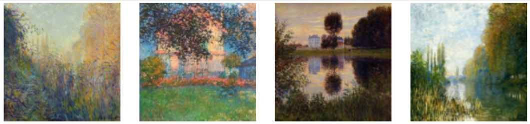
Objective: Measure the method's ability to find the atoms in a dictionary that are at the origin of several signals.

- Generate a random dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}, \quad \forall i \in [n], d_i \sim \mathcal{N}(0,\ I_K)$
- For each of the N signals that will be part of $\mathbf{Y}$:
  - Generate T_0 scalar $\forall i \in [T_0]\ \lambda_i \sim \mathcal{U}(0, 1)$
  - Choose T_0 atoms $d_{i_1}, d_{i_2}, \ldots, d_{i_{T_0}} \in \mathbf{D}$
  - Assign $\mathbf{Y_j} = \sum_{j=1}^{T_0} \lambda_j d_{i_j}$
  - Add white noise to $\mathbf{Y_j}$
- Fit the method with $\mathbf{Y}$
- Count how many atoms from $\mathbf{D}$ were recognized by the method using :

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

Image processing : Impressionist Dataset

- Impressionist paintings of size 256 × 256 pixels (Figure 3(a))
- Patches of size 8 × 8 to split images in different regions (Figure 3(b))



(a) Impressionists paintings

(b) 500 random train patches sorted by increasing variance

Figure: Training paintings (256 × 256) and 500 extracted patches (8 × 8)

Outline

1. Method
   - Sparse coding step
   - Dictionary update
   - Complexity and convergence
2. Data
   - Synthetic experiments
   - Image processing
3. Results
   - Synthetic results
   - Completion of missing pixels

Atoms recognition

We did the experiment with
$K = 50, N = 1500, n_{\text{signal}} = 20, \max_{\text{iter}} = 80$ and $T_0 = 3$ To compare this method with another we also implemented the MOD method and did the same experiment on it:

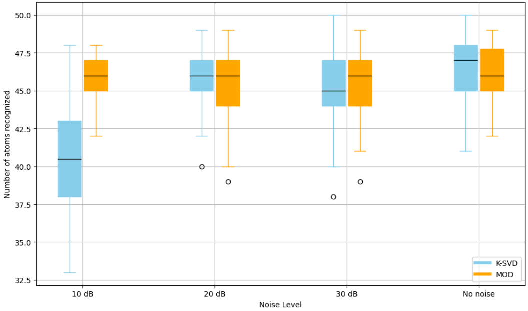
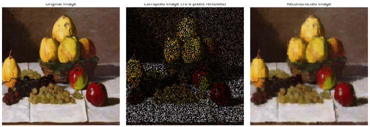


Figure: number of atoms recognized depending on the noise level

Completion of missing pixels

- Train on $N = 11000$ patches 3(b) flattened to create signals of length $n = 64$
- Dictionary of size $K = 441$
- Test on corrupted images with missing pixels

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



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

Figure: Reconstruction of *Still Life with Pears and Grapes*, Claude Monet

Conclusion

- Dictionary learning in signal reconstruction, inspired by the K-means clustering algorithm
- SVD : adapt atoms and activations at the same time while maintaining sparsity constraints
- Strong performance in recognizing dictionary atoms
- Robustness in image processing tasks (missing pixels, denoising, ...)

References I

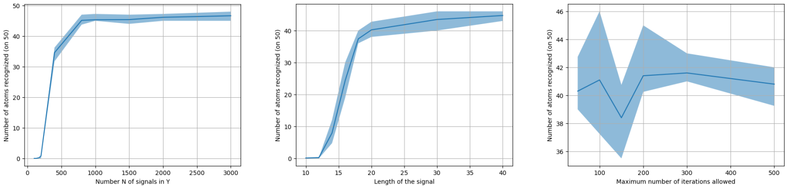
[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". In: *IEEE Transactions on Signal Processing* (2006).

[2] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition". In: *IEEE* (1993), 40–44 vol.1. DOI: 10.1109/ACSSC.1993.342465.

[3] S. A. BILLINGS S. CHEN and W. LUO. "Orthogonal least squares methods and their application to non-linear system identification". In: *International Journal of Control* 50.5 (1989), pp. 1873–1896. DOI: 10.1080/00207178908953472.

Outline

Influence of the parameters



(a) Influence of N (b) Influence of n_{signal} (c) Influence of $\max_{\text{iter}}$

Figure: Influence of some parameters

Outline

MOD method

Method of optimal direction (MOD) is also composed of two step:

- Sparse coding step: Same step as in KSVD, done with OMP in our experiment
- Dictionary update at iteration n:

$$\mathbf{D}^{(n+1)} = \mathbf{Y}\mathbf{X}^{(\mathbf{n})^T}(\mathbf{X}^{(\mathbf{n})}\mathbf{X}^{(\mathbf{n})^T})^{-1}$$

Outline

4 Influence of the parameters

5 Method of Optimal Direction

6 K-SVD Training

Learned patches from impressionist dataset

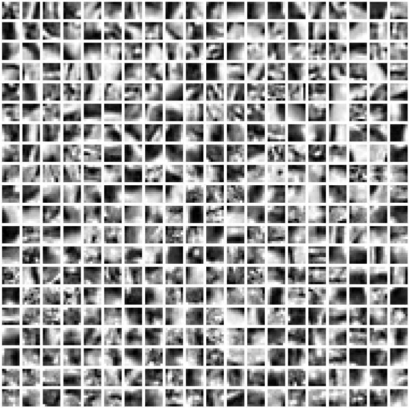
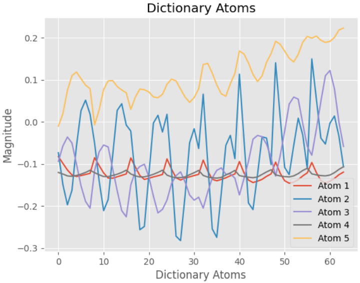
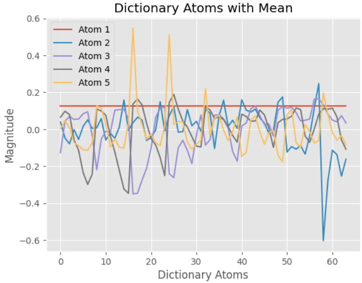


Figure: Learned patches sorted by increasing variance

K-SVD dictionary learning method comparison



(a) Simple dictionary atoms



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

Figure: First atoms of dictionaries

Error evolution during K-SVD training

Figure: Monotonic reduction in error at each stage

K-SVD dictionary learning method comparison

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

Figure: 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