Patch-Based Dictionary Learning for Sparse Image Approximation

Gerlind Plonka

Institute for Numerical and Applied Mathematics, University of Göttingen

Bologna, June 5, 2018

Outline

- Approximation model for sparse data representation
  - Minimization model
  - Dictionary learning
  - Graph regularization
- Method for dictionary learning
  - Construction of a partition tree
  - Dictionary construction
- Numerical Experiments: Seismic data denoising

Joint work with
Lina Liu and Jianwei Ma (Harbin Institute of Technology)

Approximation model for sparse data representation

Notation:

$\{\mathbf{I}_1, \ldots, \mathbf{I}_m\}$ (e.g. $\mathbf{I}_j \in \mathbb{R}^{n \times n}$), given training set of data

$\mathbf{y}_j := \mathrm{vec}\, \mathbf{I}_j \in \mathbb{R}^N$, $N = n^2$

$\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_m] \in \mathbb{R}^{N \times m}$ matrix of vectorized patches

$\mathbf{D} := [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{N \times k}$ dictionary matrix with atoms $\mathbf{d}_i \in \mathbb{R}^N$

Approximation model for sparse data representation

Notation:

$\{\mathbf{I}_1, \ldots, \mathbf{I}_m\}$ (e.g. $\mathbf{I}_j \in \mathbb{R}^{n \times n}$), given training set of data

$\mathbf{y}_j := \operatorname{vec} \mathbf{I}_j \in \mathbb{R}^N$, $N = n^2$

$\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_m] \in \mathbb{R}^{N \times m}$ matrix of vectorized patches

$\mathbf{D} := [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{N \times k}$ dictionary matrix with atoms $\mathbf{d}_i \in \mathbb{R}^N$

Sparsity promoting model:

$$\min_{\mathbf{X} \in \mathbb{R}^{k \times m}} \left(\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_0 \right),$$

$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{k \times m}$ matrix of sparse coefficient vectors

$\|\mathbf{X}\|_0$ counts the number of non-zero entries of $\mathbf{X}$

λ regularization parameter

Relaxed optimization problem

$$\min_{\mathbf{X}\in\mathbb{R}^{k\times m}} \left(\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_0 \right),$$

is "NP-hard".

Relaxed optimization problem:

$$\min_{\mathbf{X}\in\mathbb{R}^{k\times m}} \left(\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_1 \right)$$

where $\|\mathbf{X}\|_1 := \sum_{i=1}^{m} \|\mathbf{x}_i\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{k} |x_{i,j}|$.

For algorithms see e.g.
[Beck & Teboulle ('09), Needell & Vershynin ('10), Chambolle & Pock ('11)]

Model extension I: Dictionary learning

Consider

$$\min_{\mathbf{X}\in\mathbb{R}^{k\times m},\mathbf{D}\in\mathbb{R}^{N\times k}} \left(\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_*\right)$$

with $\|\cdot\|_*$ being either $\|\cdot\|_0$ or $\|\cdot\|_1$.

Dictionary learning by K-SVD

[Aharon ('06), Elad & Aharon ('06), Dong ('13)]

Idea based on alternating optimization:

1. For fixed **D** find improved **X**.
2. For fixed **X** update the dictionary **D**.

Structured dictionaries: [Cai et al. ('14), Liu et al. ('17)]

Model extension II: Graph regularization

Idea: Add a term that measures similarity between image patches

Construct a graph $G(V, E, \mathbf{W})$ with $V = \{\mathbf{I}_1, \ldots, \mathbf{I}_m\}$.
$\mathbf{I}_i$, $\mathbf{I}_j$ are connected by an edge with weight $W_{i,j}$.

Model extension II: Graph regularization

Idea: Add a term that measures similarity between image patches

Construct a graph $G(V, E, \mathbf{W})$ with $V = \{\mathbf{I}_1, \dots, \mathbf{I}_m\}$.
$\mathbf{I}_i$, $\mathbf{I}_j$ are connected by an edge with weight $W_{i,j}$.

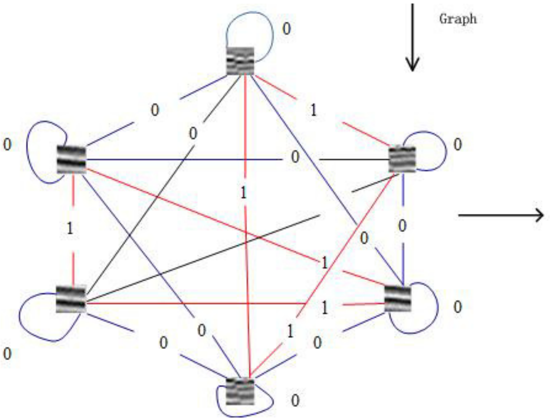
Choice of the weight matrix $\mathbf{W} = (W_{i,j})_{i,j=1}^m \in \mathbb{R}^{m \times m}$:

1. Find the K nearest neighbors of $\mathbf{I}_j$ by inspecting theses distances $\|\mathbf{I}_i - \mathbf{I}_k\|_F^2$ for $k \in \{1, \dots, i-1, i+1, \dots, m\}$.
2. Define the symmetric weight matrix $\mathbf{W} = (W_{i,j})_{i,j=1}^m$ by

$$W_{i,j} = \begin{cases} 1 & \text{if } \mathbf{I}_j \text{ is among the } K \text{ nearest neighbors of } \mathbf{I}_i \\ & \text{or } \mathbf{I}_i \text{ is among the } K \text{ nearest neighbors of } \mathbf{I}_j \\ 0 & \text{otherwise.} \end{cases}$$

3. Introduce $\Delta = \mathrm{diag}(\Delta_1, \dots, \Delta_m) \in \mathbb{R}^{m \times m}$ with $\Delta_i = \sum_{j=1}^m W_{i,j}$.
4. Define the Laplacian matrix of the graph G: $\mathbf{L} = \Delta - \mathbf{W} \in \mathbb{R}^{m \times m}$.

Model extension II: Graph regularization



$$W = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Model extension II: Graph regularization

Then

$$\mathrm{Tr}(\mathbf{YLY}^T) = \sum_{i,j=1}^{m} W_{i,j}\|\mathbf{I}_i - \mathbf{I}_j\|_F^2 = \sum_{i,j=1}^{m} W_{i,j}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2 = \sum_{\mathbf{I}_i\sim\mathbf{I}_j} \|\mathbf{I}_i - \mathbf{I}_j\|_F^2$$

where $\mathbf{I}_i \sim \mathbf{I}_j$ if $W_{i,j} = 1$.

We suppose that the dictionary atoms $\mathbf{x}_j$, $j = 1, \ldots, m$ possess a similar topological structure as $\mathbf{y}_j$, $j = 1, \ldots, m$, and introduce

$$\mathrm{Tr}(\mathbf{XLX}^T) = \sum_{i,j=1}^{m} W_{i,j}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \sum_{\mathbf{I}_i\sim\mathbf{I}_j} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$$

Model generalization:

$$\min_{\mathbf{X}\in\mathbb{R}^{k\times m}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \alpha\,\mathrm{Tr}(\mathbf{XLX}^T) + \lambda\|\mathbf{X}\|_1 \qquad \alpha \geq 0.$$

See also Yankelevsky & Elad (2016).

Method for dictionary learning

Dictionary construction.

1. Construct of a partition tree (similar to [ZENG ('15)])
2. Determine the dictionary from the partition tree

Step 1. Construct of a partition tree

- Compute the mean of all training patches

$$\mathbf{C} := \frac{1}{m} \sum_{i=1}^{m} \mathbf{I}_i \in \mathbb{R}^{n \times n}$$

and the covariance matrices

$$\mathbf{C}_L := \frac{1}{m} \sum_{i=1}^{m} (\mathbf{I}_i - \mathbf{C})(\mathbf{I}_i - \mathbf{C})^T, \quad \mathbf{C}_R := \frac{1}{m} \sum_{i=1}^{m} (\mathbf{I}_i - \mathbf{C})^T (\mathbf{I}_i - \mathbf{C}).$$

Construct a partition tree

- Compute the normalized eigenvectors $\mathbf{u}$ and $\mathbf{v}$

$$\mathbf{u} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_L \mathbf{x}, \qquad \mathbf{v} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_R \mathbf{x},$$

representing the main structures of the training patches not being captured by the mean patch $\mathbf{C}$.

Construct a partition tree

- Compute the normalized eigenvectors $\mathbf{u}$ and $\mathbf{v}$

$$\mathbf{u} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_L \mathbf{x}, \qquad \mathbf{v} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_R \mathbf{x},$$

  representing the main structures of the training patches not being captured by the mean patch $\mathbf{C}$.
- Compute $s_i := \mathbf{u}^T \mathbf{I}_i \mathbf{v}$, $i = 1, \ldots, m$, and order these numbers by size, $s_{\ell_1} \leq s_{\ell_2} \leq \ldots \leq s_{\ell_m}$.

Construct a partition tree

- Compute the normalized eigenvectors $\mathbf{u}$ and $\mathbf{v}$

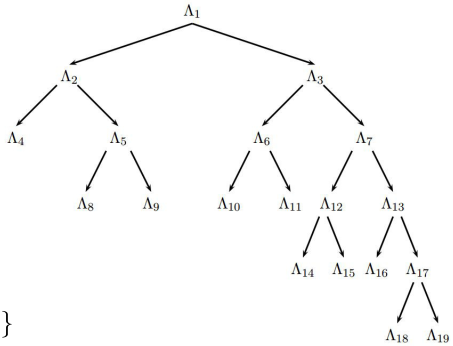
$$\mathbf{u} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_L \mathbf{x}, \qquad \mathbf{v} := \operatorname*{argmax}_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{C}_R \mathbf{x},$$

  representing the main structures of the training patches not being captured by the mean patch $\mathbf{C}$.
- Compute $s_i := \mathbf{u}^T \mathbf{I}_i \mathbf{v}$, $i = 1, \ldots, m$, and order these numbers by size, $s_{\ell_1} \leq s_{\ell_2} \leq \ldots \leq s_{\ell_m}$.
- Compute

$$\hat{\kappa} := \operatorname*{argmin}_{1 \leq \kappa \leq m-1} \left[\sum_{r=1}^{\kappa} \left(s_{\ell_r} - \frac{1}{\kappa} \sum_{\nu=1}^{\kappa} s_{\ell_\nu} \right)^2 + \sum_{r=\kappa+1}^{m} \left(s_{\ell_r} - \frac{1}{m-\kappa} \sum_{\nu=\kappa+1}^{m} s_{\ell_\nu} \right)^2 \right].$$

  to derive the partition $\{\mathbf{I}_{\ell_1}, \ldots, \mathbf{I}_{\ell_{\hat{\kappa}}}\} \cup \{\mathbf{I}_{\ell_{\hat{\kappa}+1}}, \ldots, \mathbf{I}_{\ell_m}\}$.
- Partition the two obtained subsets further using the same scheme.

Example of a partition tree



$$
\begin{aligned}
\Lambda_1 &= \{1, \ldots, 10\} \\
\Lambda_2 &= \{5, 8, 9\}, \quad \Lambda_3 = \{1, 2, 3, 4, 6, 7, 10\}, \\
\Lambda_4 &= \{8\}, \quad \Lambda_5 = \{5, 9\}, \quad \Lambda_6 = \{3, 6\}, \quad \Lambda_7 = \{1, 2, 4, 7, 10\} \\
\Lambda_8 &= \{5\}, \quad \Lambda_9 = \{9\}, \quad \Lambda_{10} = \{3\}, \ \Lambda_{11} = \{6\}, \ \Lambda_{12} = \{1, 7\}, \ \Lambda_{13} = \{2, 4, 10\} \\
\Lambda_{14} &= \{1\}, \quad \Lambda_{15} = \{7\}, \quad \Lambda_{16} = \{2\}, \quad \Lambda_{17} = \{4, 10\}, \\
\Lambda_{18} &= \{4\}, \quad \Lambda_{19} = \{0\}.
\end{aligned}
$$

Determine the dictionary from the partition tree

Each node in the tree is associated with a subset $\{\mathbf{I}_j\}_{j \in \Lambda_k}$.
For each index set Λ_k, compute

$$\mathbf{C}_k := \frac{1}{|\Lambda_k|} \sum_{i \in \Lambda_k} \mathbf{I}_i$$

and

$$\mathbf{u}_k := \operatorname*{argmax}_{\|\mathbf{x}\|_2 = 1} \mathbf{C}_k \mathbf{C}_k^T \mathbf{x}, \qquad \mathbf{v}_k := \operatorname*{argmax}_{\|\mathbf{x}\|_2 = 1} \mathbf{C}_k^T \mathbf{C}_k \mathbf{x}.$$

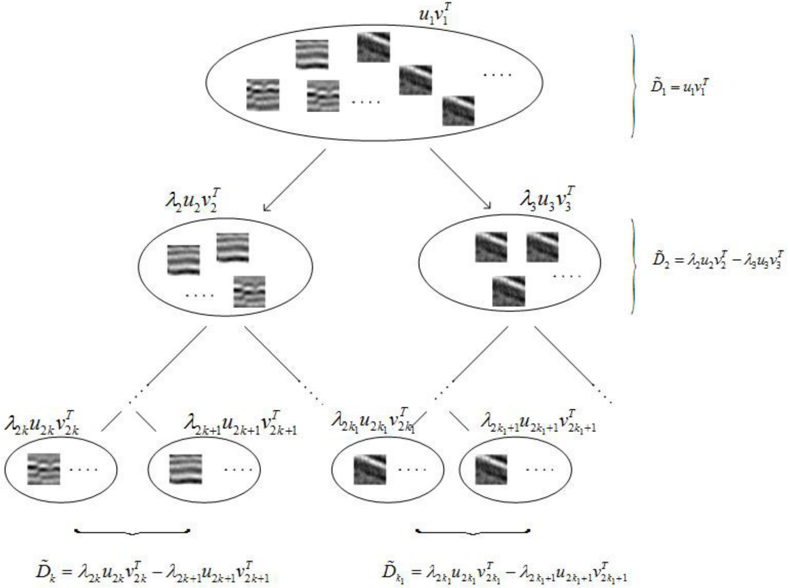
First dictionary element:

$$\mathbf{D}_1 := \mathbf{u}_1 \mathbf{v}_1^T$$

Further dictionary elements: For each pair of children nodes with index sets Λ_{2k}, Λ_{2k+1} and center matrices $\mathbf{C}_{2k}$, $\mathbf{C}_{2k+1}$ let

$$\tilde{\mathbf{D}}_k := \lambda_{2k} \mathbf{u}_{2k} \mathbf{v}_{2k}^T - \lambda_{2k+1} \mathbf{u}_{2k+1} \mathbf{v}_{2k+1}^T, \qquad \mathbf{D}_k := \frac{\tilde{\mathbf{D}}_k}{\|\tilde{\mathbf{D}}_k\|_F},$$

Determine the dictionary from the partition tree



Application for denoising

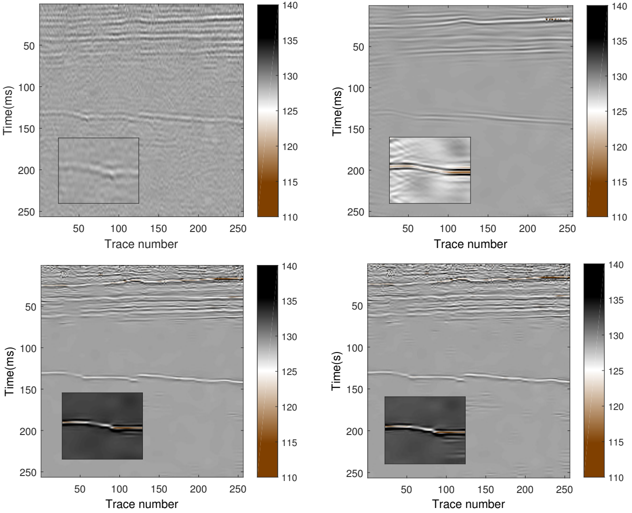
Denoising algorithm with dictionary learning and graph regularization

Input: Noisy training data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$
Number of iterations
Parameters K, α and λ

1. Set $\mathbf{Y}_D := \mathbf{Y}$. Loop through steps 2-5 until the given number of iterations is achieved:
2. Compute the Laplacian matrix $\mathbf{L}$ for the given training set $\mathbf{Y}_D$.
3. Determine the dictionary $\mathbf{D}$ by a dictionary learning algorithm based on $\mathbf{Y}_D$.
4. Solve the minimization problem
$$\min_{\mathbf{X} \in \mathbb{R}^{k \times m}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \alpha \operatorname{Tr}(\mathbf{XLX}^T) + \lambda \|\mathbf{X}\|_1.$$
5. Reconstruct the data $\mathbf{Y}_D := \mathbf{DX}$.

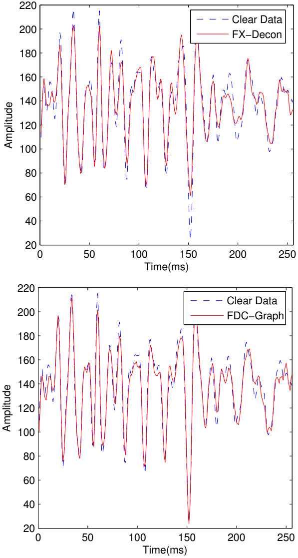
Output: Denoised data $\mathbf{Y}_D$.

Denoising results for field data



Denoising using FX-Decon, Curvelets, FDC-Graph and SDC-Graph

Denoising results: Single trace comparison



Summary

- We have considered a generalized model

$$\min_{\mathbf{X}\in\mathbb{R}^{k\times m}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \alpha\,\mathrm{Tr}(\mathbf{X}\mathbf{L}\mathbf{X}^T) + \lambda\|\mathbf{X}\|_1 \qquad \alpha \geq 0.$$

  including a learned dictionary and a graph regularization term.
- Dictionary learning is based on a partition tree.
- The partition of training patches uses SVD of patches.
- This method exploits two-dimensional geometric structure of the training data.
- The dictionary learning method is essentially cheaper than K-SVD.
- See the talk by Renato Budinich: Clustering based dictionary learning, Thursday, 9:50, CP6.

References

- Lina Liu, Jianwei Ma, and Gerlind Plonka
  Sparse graph-regularized dictionary learning for suppressing random seismic noise.
  Geophysics 83(3) (2018), V215–V231.

- Lina Liu, Gerlind Plonka, and Jianwei Ma
  Seismic data interpolation and denoising by learning a tensor tight frame.
  Inverse Problems 33(10) (2017), 105011.

\thankyou