Structured Support Exploration For Multilayer Sparse Matrix Factorization

Quoc Tung Le   Rémi Gribonval

Université de Lyon, Ecole Normale Supérieure de Lyon

INRIA, CNRS, Laboratoire du Parallélisme

ICASSP 6 - 11 June 2021

Multilayer sparse matrix factorization problem

- Given a matrix A, we would like to find J matrices $S_1, S_2, \ldots, S_J$ satisfying

$$A \approx S_1 S_2 \ldots S_J$$

such that $S_1, S_2, \ldots S_J$ are sparse matrices.

Motivation: Faster linear operator

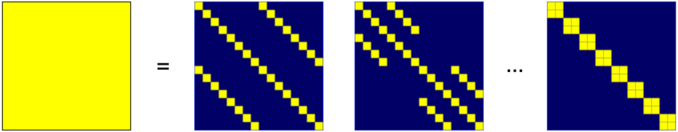
- Given a matrix $A \approx S_1 S_2 \dots S_J$ and a vector x, then
$$Ax \approx S_1(S_2(\dots(S_J x))) \tag{1}$$

Motivation: Faster linear operator

- Given a matrix $A \approx S_1 S_2 \dots S_J$ and a vector x, then
$$Ax \approx S_1(S_2(\dots(S_J x))) \quad (1)$$
- If $S_1, S_2, \dots, S_J$ are **sparse** matrices, evaluating the RHS of equation (1) must be faster than directly evaluating the LHS.

Motivation: Faster linear operator

- Given a matrix $A \approx S_1 S_2 \dots S_J$ and a vector x, then
$$Ax \approx S_1(S_2(\dots(S_J x))) \quad (1)$$
- If $S_1, S_2, \dots, S_J$ are **sparse** matrices, evaluating the RHS of equation (1) must be faster than directly evaluating the LHS.
- Application: the Fast Fourier Transform, the Fast Hadamard Transform, etc.



The factorization of the Discrete Fourier Transform.

Problem formulation

Sparse Factorization Problem Formulation

Given a matrix A, $J \in \mathbb{N}$ and $\delta_{\mathcal{E}_j}$ as indicator function of set $\mathcal{E}_j$, solve:

$$\underset{S_1,\ldots,S_J}{\text{Minimize}} \quad \underbrace{\|A - \prod_{j=1}^{J} S_j\|_F^2}_{\text{Data fidelity}} + \underbrace{\sum_{j=1}^{J} \delta_{\mathcal{E}_j}(S_j)}_{\text{Sparsity-inducing penalty}} \tag{2}$$

Problem formulation

Sparse Factorization Problem Formulation

Given a matrix A, $J \in \mathbb{N}$ and $\delta_{\mathcal{E}_j}$ as indicator function of set $\mathcal{E}_j$, solve:

$$\underset{S_1,\ldots,S_J}{\text{Minimize}} \ \underbrace{\|A - \prod_{j=1}^{J} S_j\|_F^2}_{\text{Data fidelity}} + \underbrace{\sum_{j=1}^{J} \delta_{\mathcal{E}_j}(S_j)}_{\text{Sparsity-inducing penalty}} \qquad (2)$$

Choice of matrices set $\mathcal{E}$:

- $\mathcal{E}^k_{row} = \{S \mid \|S_{i,\bullet}\|_0 \leq k\}$: at most k nonzero entries per row.
- $\mathcal{E}^k_{col} = \{S \mid \|S_{\bullet,i}\|_0 \leq k\}$: at most k nonzero entries per column.
- $\mathcal{E}^k_{tot} = \{S \mid \|S\|_0 \leq k\}$: at most k nonzero entries in total.

Problem Approach - State of The Art

- MAIN TOOLS: proximal operator and proximal algorithm[1].

[1] Neal Parikh and Stephen Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), 127–239. ISSN: 2167-3888. DOI: 10.1561/2400000003.

Problem Approach - State of The Art

- MAIN TOOLS: proximal operator and proximal algorithm[1].
- STATE OF THE ART: Proximal Alternating Linearized Minimization (PALM)[2], and its applications in matrix factorization.[3].

[1] Neal Parikh and Stephen Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), 127–239. ISSN: 2167-3888. DOI: 10.1561/2400000003.

[2] Jerome Bolte, Shoham Sabach, and Marc Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1-2 (2014), pp. 459–494.

[3] Luc Le Magoarou and Rémi Gribonval. "Flexible Multi-layer Sparse Approximations of Matrices and Applications". In: *IEEE Journal of Selected Topics in Signal Processing* 10.4 (June 2016), pp. 688–700.

Problem Approach - State of The Art

- MAIN TOOLS: proximal operator and proximal algorithm[1].
- STATE OF THE ART: Proximal Alternating Linearized Minimization (PALM)[2], and its applications in matrix factorization.[3].
- PALM IN A NUTSHELL:

1: **for** $j \in \{1, \ldots, J\}$ **do**

2: $$S_j^{i+1} \leftarrow P_{\mathcal{E}_j}\underbrace{\left(S_j^i - \frac{1}{c_j^i}\nabla_{S_j}\|A - \lambda(\prod_{l=1}^{j-1} S_l^{i+1})(\prod_{l=j}^{J} S_l^i)\|_F^2\right)}_{\text{gradient step}}$$

3: **end for**

[1] Neal Parikh and Stephen Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), 127–239. ISSN: 2167-3888. DOI: 10.1561/2400000003.

[2] Jerome Bolte, Shoham Sabach, and Marc Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1-2 (2014), pp. 459–494.

[3] Luc Le Magoarou and Rémi Gribonval. "Flexible Multi-layer Sparse Approximations of Matrices and Applications". In: *IEEE Journal of Selected Topics in Signal Processing* 10.4 (June 2016), pp. 688–700.

Problems with PALM

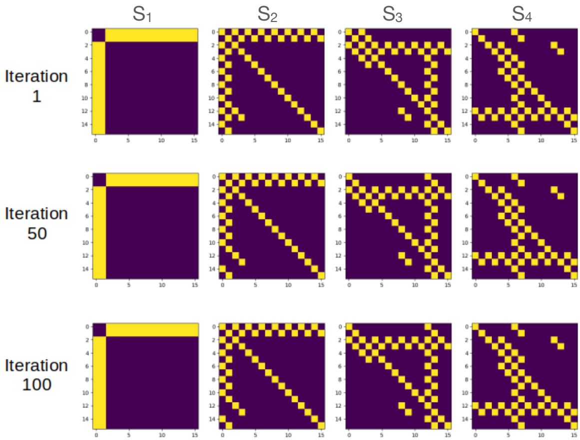
- PROBLEM 1: PALM is **lazy** at exploring support.

  CONTRIBUTION 1: Enforce the search of support with a new algorithm.

- PROBLEM 2: Proximal operators of existing sparse matrix sets often produce **rank deficient** factors.

  CONTRIBUTION 2: Propose a new family of sparse matrices to avoid rank deficiency.

Support Exploration Laziness and Rank Deficiency



The evolution of support of factors of PALM during the Hadamard Transform factorization. Yellow indicates the support.

Bilinear Hard Thresholding Pursuit (BHTP)

- Consider 2-factor factorization: $A = S_1 S_2 = XY, X \in \mathcal{E}_X, Y \in \mathcal{E}_Y$.

[4] Simon Foucart. "Hard Thresholding Pursuit: An Algorithm for Compressive Sensing". In: *SIAM J. Numerical Analysis* 49 (Jan. 2011), pp. 2543–2563.

Bilinear Hard Thresholding Pursuit (BHTP)

- Consider 2-factor factorization: $A = S_1 S_2 = XY, X \in \mathcal{E}_X, Y \in \mathcal{E}_Y$.
- If X is known, finding Y is a linear inverse problem.

[4] Simon Foucart. "Hard Thresholding Pursuit: An Algorithm for Compressive Sensing". In: *SIAM J. Numerical Analysis* 49 (Jan. 2011), pp. 2543–2563.

Bilinear Hard Thresholding Pursuit (BHTP)

- Consider 2-factor factorization: $A = S_1 S_2 = XY, X \in \mathcal{E}_X, Y \in \mathcal{E}_Y$.
- If X is known, finding Y is a linear inverse problem.
- Proposal: Using Hard Thresholding Pursuit[4].

[4] Simon Foucart. "Hard Thresholding Pursuit: An Algorithm for Compressive Sensing". In: *SIAM J. Numerical Analysis* 49 (Jan. 2011), pp. 2543–2563.

Bilinear Hard Thresholding Pursuit (BHTP)

- Consider 2-factor factorization: $A = S_1 S_2 = XY, X \in \mathcal{E}_X, Y \in \mathcal{E}_Y$.
- If X is known, finding Y is a linear inverse problem.
- Proposal: Using Hard Thresholding Pursuit[4].

$Y_0 = 0$.

for $n \in \{1, \dots, N\}$ **do**

$$T_n = \operatorname{supp}\left(P_{\mathcal{E}_Y} \underbrace{\left(Y_{n-1} - \lambda \nabla_Y \|A - XY_{n-1}\|^2\right)}_{\text{gradient step}} \right).$$

$$Y_n = \arg\min_Y \|A - XY\|, \operatorname{supp}(Y) \subseteq T_n$$ (**orthogonal projection**).[5]

end for

[4] Simon Foucart. "Hard Thresholding Pursuit: An Algorithm for Compressive Sensing". In: *SIAM J. Numerical Analysis* 49 (Jan. 2011), pp. 2543–2563.

[5] Can be replaced by multiple gradient steps to reduce calculating time

Bilinear Hard Thresholding Pursuit (BHTP)

- Consider 2-factor factorization: $A = S_1 S_2 = XY, X \in \mathcal{E}_X, Y \in \mathcal{E}_Y$.
- If X is known, finding Y is a linear inverse problem.
- Proposal: Using Hard Thresholding Pursuit[4].

1: $Y_0 = 0$.
2: **for** $n \in \{1, \ldots, N\}$ **do**
3: $$T_n = \text{supp}\left(P_{\mathcal{E}_Y}\underbrace{\left(Y_{n-1} - \lambda \nabla_Y \|A - XY_{n-1}\|^2\right)}_{\text{gradient step}}\right).$$
4: $Y_n = \arg\min_Y \|A - XY\|, \text{supp}(Y) \subseteq T_n$ (**orthogonal projection**).[5]
5: **end for**

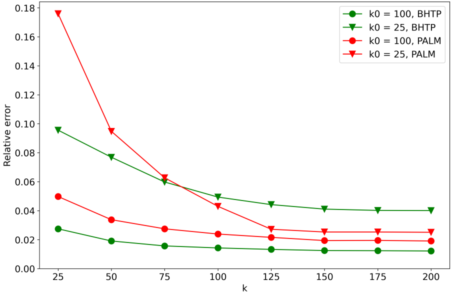
- Alternating between two factors (Bilinear Hard Thresholding Pursuit).

[4] Simon Foucart. "Hard Thresholding Pursuit: An Algorithm for Compressive Sensing". In: *SIAM J. Numerical Analysis* 49 (Jan. 2011), pp. 2543–2563.

[5] Can be replaced by multiple gradient steps to reduce calculating time

Experimental Results

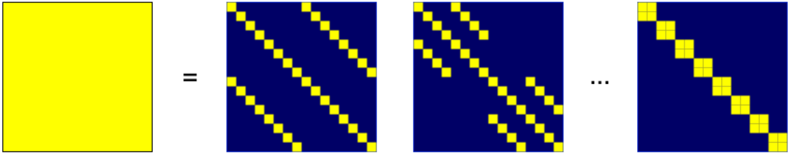
- Factorize a leadfield matrix MEG, (in functional brain imaging), $A \in \mathbb{R}^{8193\times 204}$.
- Constraints: $\mathcal{E}_X = \mathcal{E}^{k_0}_{col}, \mathcal{E}_Y = \mathcal{E}^{k}_{row}$.
- Measure: Relative error $= \|A - XY\|_F / \|A\|_F$.



The error of BHTP and PALM factorizing the matrix MEG.

Motivation

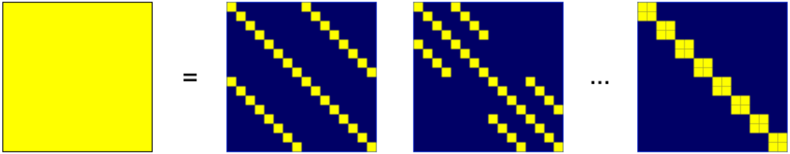
- Factors of the Hadamard Transform (and the Discrete Fourier Transform) have 2 nonzero entries per row and column.



The factorization of the Discrete Fourier Transform.

Motivation

- Factors of the Hadamard Transform (and the Discrete Fourier Transform) have 2 nonzero entries per row and column.



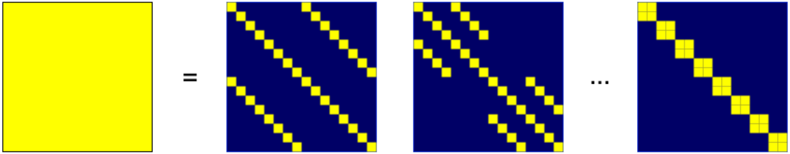
The factorization of the Discrete Fourier Transform.

Definition

A k-regular sparse matrix $U \in \mathbb{C}^{n \times n}$ is a matrix whose columns and rows contain at most k non-zero entries each.
Let $\mathcal{R}_k$ be the set of all k-regular matrices.

Motivation

- Factors of the Hadamard Transform (and the Discrete Fourier Transform) have 2 nonzero entries per row and column.



The factorization of the Discrete Fourier Transform.

Definition

A k-regular sparse matrix $U \in \mathbb{C}^{n\times n}$ is a matrix whose columns and rows contain at most k non-zero entries each.
Let $\mathcal{R}_k$ be the set of all k-regular matrices.

Problem: How to project a matrix onto $\mathcal{R}_k$.

Generalized Hungarian Method (GHM)

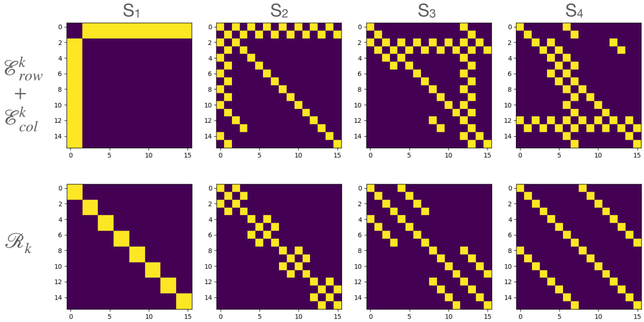
- Proximal operator corresponding to $\mathcal{R}_k$ can be reduced to a bipartite graph problem. Our proposed algorithm generalizes the **Hungarian Method** (hence the name **Generalized Hungarian Method**).

Generalized Hungarian Method (GHM)

- Proximal operator corresponding to $\mathcal{R}_k$ can be reduced to a bipartite graph problem. Our proposed algorithm generalizes the **Hungarian Method** (hence the name **Generalized Hungarian Method**).
- The complexity is $O(kn^3)$.

Generalized Hungarian Method (GHM)

- Proximal operator corresponding to $\mathcal{R}_k$ can be reduced to a bipartite graph problem. Our proposed algorithm generalizes the **Hungarian Method** (hence the name **Generalized Hungarian Method**).
- The complexity is $O(kn^3)$.



The result of PALM with different proximal operators.

Conclusion

Highlight of the paper:

- Analysis of the laziness of PALM in exploring the support. This holds with many other proximal algorithms as well
  $\rightarrow$ *Contribution 1: Bilinear Hard Thresholding Pursuit.*
- Rank deficient factors produced by existing proximal operator.
  $\rightarrow$ *Contribution 2: k-regular sparse matrix and its proximal operator.*

Conclusion

Highlight of the paper:

- Analysis of the laziness of PALM in exploring the support. This holds with many other proximal algorithms as well
  → *Contribution 1: Bilinear Hard Thresholding Pursuit.*
- Rank deficient factors produced by existing proximal operator.
  → *Contribution 2: k-regular sparse matrix and its proximal operator.*

What is left for future work:

- Other operators (such as the Discrete Fourier Transform (DFT)) resist the current algorithms.
  → Possible causes: Complex value optimization, the DFT admits fewer exact factorizations.
- Extension for nonlinear setting (Neural Networks, etc).

Practical implementation - Pyfaust

- Visit faust.inria.fr.
- Available for python with "pip install pyfaust" and also for Matlab.
- Includes GPU accelerations

Practical implementation - Pyfaust

- Visit faust.inria.fr.
- Available for python with "pip install pyfaust" and also for Matlab.
- Includes GPU accelerations

Linearization

Primal for Lorenz component L_k

$$
\begin{aligned}
\min\ & kr_k - \sum_{i=1}^{n} d_{ki} \\
\text{s.t. } & r_k - y_i && \geq && d_{ki} && \forall i \in [n], \forall k \in [n] \\
& d_{ki} && \geq && Mz_{ki} && \forall i \in [n], \forall k \in [n] \\
& \sum_{i=1}^{n} z_{ki} && \leq && k-1 && \forall k \in [n] \\
& d_{ki} && \leq && 0 && \forall i \in [n], \forall k \in [n].
\end{aligned} \tag{3}
$$