Revisiting CHAMPAGNE: Sparse Bayesian Learning as Reweighted Sparse Coding

Dylan Sechet, Matthieu Kowalski
Laboratoire Interdisciplinaire des Sciences du Numériques
Inria, Université Paris-Saclay, CNRS
Gif-sur-Yvette, France
{dylan.sechet,matthieu.kowalski}@inria.fr

Samy Mokhtari, Bruno Torrésani
Aix Marseille Univ, CNRS, I2M
Marseille, France
{samy.mokhtari, bruno.torresani}@univ-amu.fr

***Abstract*—This paper revisits the CHAMPAGNE algorithm within the Sparse Bayesian Learning (SBL) framework and establishes its connection to reweighted sparse coding. We demonstrate that the SBL objective can be reformulated as a reweighted ℓ_{21}-minimization problem, providing a more straightforward interpretation of the sparsity mechanism and enabling the design of an efficient iterative algorithm. Additionally, we analyze the behavior of this reformulation in the low signal-to-noise ratio (SNR) regime, showing that it simplifies to a weighted ℓ_{21}-regularized least squares problem. Numerical experiments validate the proposed approach, highlighting its improved computational efficiency and ability to produce exact sparse solutions, particularly in simulated MEG source localization tasks.**

***Index Terms*—Inverse problem, Sparse Bayesian Learning, reweighted (group)-lasso**

I. Introduction

Inverse problems focus on reconstructing unknown parameters or sources from incomplete and noisy observed data. In the context of magnetoencephalography (MEG), this involves estimating cortical activity from sensors located at several points around the subject's head [1]. Such an inverse problem is highly under-determined, where the number of potential sources exceeds the available measurements. Addressing this under-determination requires introducing some regularization techniques or prior information.

Among these strategies, the minimum norm estimate (MNE) [2] is a classical approach that favors smooth solutions by minimizing the ℓ_2 norm of the sources. However, MNE fails to capture the localized nature of brain activity. Sparsity-promoting methods have been developed to address this limitation, leveraging the assumption that brain activity is spatially sparse.

In particular, lasso-type methods [3], [4], which include ℓ_1-regularized least squares and extensions such as the group lasso [5], enforce sparsity directly on the solution, and have been widely adopted for MEG source localization [6], [7]. They use convex penalties, and their optimization problems can be efficiently solved with iterative algorithms, such as the (Fast) Iterative Shrinkage-Thresholding Algorithm ((F)ISTA) [8], [9]. However, lasso-based methods rely heavily on manually tuning regularization parameters, which can be challenging in practice.

In contrast, Bayesian frameworks such as Sparse Bayesian Learning (SBL) [10] require minimal tuning, as the only parameter to adjust is the noise variance, which can also be estimated within the framework. SBL introduces a hierarchical Bayesian model where source variances act as hyperparameters driving sparsity, and solutions are estimated iteratively. Among SBL algorithms, *CHAMPAGNE* [11], [12] has been particularly successful, offering robust solutions by jointly estimating the sources and their variances. However, while CHAMPAGNE induces sparsity in source variances, the corresponding source estimates remain only approximately sparse, with small but not exactly zero components. This introduces the need for manual thresholding.

a) Contributions and outline: In this work, we revisit the CHAMPAGNE algorithm within the SBL framework by reformulating it as a reweighted sparse coding problem. Section II introduces the general SBL model and reviews the CHAMPAGNE algorithm. Section III demonstrates that the SBL objective can be expressed as a reweighted sparse coding problem, which leads to a more efficient algorithm. We also provide a convergence analysis. Furthermore, we examine the behavior of this reformulation in the low signal-to-noise ratio (SNR) regime. Finally, our numerical experiments validate our approach, showing improved computational efficiency and achieving exact sparsity in the solutions.

b) Notations: Vectors are in bold lowercase (e.g., $\mathbf{v}$), and matrices in bold uppercase (e.g., $\mathbf{M}$). The element at the i-th row and j-th column of a matrix $\mathbf{M}$ is denoted by $M[i,j]$, while $\mathbf{M}[i,:]$ represents the i-th row, and $\mathbf{M}[:,j]$ denotes the j-th column. Similarly, the i-th coordinate of a vector $\mathbf{v}$ is denoted by $v[i]$ or v_i

II. SBL and CHAMPAGNE algorithm

This section presents the general SBL model. For the sake of simplicity, we assume that the noise variance is known. We then focus on the CHAMPAGNE algorithm, which adopts a majorize-minimize strategy relying on a convex upper bound of the SBL objective.

This work was supported by the French National Agency for Research through the BMWs project (ANR-20-CE45-0018)

A. SBL model

In the context of MEG, the objective is to estimate the cortical activity $\mathbf{X}$ from the sensor measurements $\mathbf{Y}$. This relationship is modeled as follows:

$$\mathbf{Y} = \mathbf{G}\mathbf{X} + \mathbf{N}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times T}$ represents the observed sensor data, $\mathbf{G} \in \mathbb{R}^{M\times N}$ is the leadfield matrix describing the mapping from cortical sources to sensors, $\mathbf{X} \in \mathbb{R}^{N\times T}$ denotes the unknown source activities, and $\mathbf{N} \in \mathbb{R}^{M\times T}$ is additive white (in space and time) Gaussian noise with $N[n,t] \sim \mathcal{N}(0, \sigma_e^2), \forall n, t.$.

To promote sparsity, SBL models the source activity $\mathbf{X}$ as a zero-mean multivariate Gaussian with a diagonal covariance matrix $\mathbf{\Gamma}$:

$$\mathbf{X}[:,t] \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}), \quad \mathbf{\Gamma} = \mathrm{Diag}\left[\boldsymbol{\gamma}\right], \quad \forall n, \gamma_n \geq 0, \tag{2}$$

where the hyperparameters γ_n encode the variance of each source, indirectly controlling its sparsity. While various priors can be placed on γ_n, we focus on an exponential prior:

$$\gamma_n \sim \mathcal{E}(\rho), \tag{3}$$

a particular case of the classical Gamma prior often used for scale parameters.

To estimate $\boldsymbol{\gamma}$, SBL proposes to maximize the marginal likelihood of the data (type-II likelihood), which leads to minimizing the following cost function:

$$\begin{aligned}\mathcal{L}^{II}(\boldsymbol{\gamma}) &= -\log p(\mathbf{Y}, \boldsymbol{\gamma}) = -\log p(\mathbf{Y}|\boldsymbol{\gamma}) - \log p(\boldsymbol{\gamma}) \\ &= \frac{1}{2}\mathbf{Y}^T\mathbf{\Sigma}_y(\boldsymbol{\gamma})^{-1}\mathbf{Y} + \frac{1}{2}\log|\mathbf{\Sigma}_y(\boldsymbol{\gamma})| + \rho\sum_{n=1}^{N}\gamma_n,\end{aligned} \tag{4}$$

where:

$$\mathbf{\Sigma}_y(\boldsymbol{\gamma}) = \sigma_e^2\mathbf{I}_M + \mathbf{G}\mathbf{\Gamma}\mathbf{G}^T. \tag{5}$$

To optimize $\mathcal{L}^{II}(\boldsymbol{\gamma})$, the Expectation-Maximization (EM) algorithm [10] alternates updates of the sources $\mathbf{X}$ and the hyperparameters $\boldsymbol{\gamma}$. However, it tends to converge slowly. In contrast, the CHAMPAGNE algorithm [11] utilizes a convex bounding strategy, resulting in significantly faster computations. Hashemi et al. [12] provides a unified perspective on these and related SBL algorithms, highlighting CHAMPAGNE as an effective and scalable solution, which is further elaborated in the next section.

B. CHAMPAGNE algorithm

The CHAMPAGNE algorithm comes from reformulating the original marginal likelihood objective as a minimization problem involving the sources $\mathbf{X}$, the hyperparameters $\boldsymbol{\gamma}$, and an auxiliary dual variable $\mathbf{v}$. Th e resulting cost function is expressed as:

$$\begin{aligned}\mathcal{C}(\mathbf{X}, \boldsymbol{\gamma}, \mathbf{v}) =& \frac{1}{T\sigma_e^2}\|\mathbf{Y} - \mathbf{G}\mathbf{X}\|^2 + \frac{1}{T}\sum_{n=1}^{N}\sum_{t=1}^{T}\frac{X[n,t]^2}{\gamma_n} \\ &+ \mathbf{v}^T\boldsymbol{\gamma} - w^*(\mathbf{v}) + \rho\sum_{n=1}^{N}\gamma_n,\end{aligned} \tag{6}$$

where $w^*(\mathbf{v})$ is the concave conjugate of $\log|\mathbf{\Sigma}_y(\boldsymbol{\gamma})|$:

$$w^*(\mathbf{v}) = \max_{\boldsymbol{\gamma}} \mathbf{v}^T\boldsymbol{\gamma} - \log|\mathbf{\Sigma}_y(\boldsymbol{\gamma})|, \tag{7}$$

The CHAMPAGNE algorithm alternates between updates of $\mathbf{v}$, $\mathbf{X}$, and $\boldsymbol{\gamma}$, minimizing Eq. (6) with respect to each variable while keeping the others fixed. The updates are computed iteratively as follows:

$$\mathbf{v}^{(i)} = \mathrm{Diag}\left[\mathbf{G}^T\mathbf{\Sigma}_y^{-1}(\boldsymbol{\gamma}^{(i-1)})\mathbf{G}\right], \tag{8}$$

$$\mathbf{X}^{(i)} = \mathbf{\Gamma}^{(i-1)}\mathbf{G}^T\mathbf{\Sigma}_y^{-1}(\boldsymbol{\gamma}^{(i-1)})\mathbf{Y}, \tag{9}$$

$$\gamma_n^{(i)} = \frac{\|\mathbf{X}^{(i)}[n,:]\|}{\sqrt{T(v_n + \rho)}}. \tag{10}$$

The iterative updates have been shown to converge to a local minimum [12].

III. SBL BY (RE)WEIGHTED SPARSE CODING

This section shows that the original SBL optimization problem can be reformulated as a reweighted sparse coding problem, where sparsity is enforced iteratively through adaptive weights. Then, we analyze the behavior of this reformulation in the low SNR regime, where the log-determinant term simplifies, leading to a weighted ℓ_{21}-regularized least squares problem.

A. Reformulation and Algorithm

The following proposition shows that for a fixed $\mathbf{v}$, the minimization of Eq. (6) reduces to a weighted (group)-lasso sparse coding problem.

Proposition 1. *Minimizing the joint cost function concerning both $\mathbf{X}$ and $\boldsymbol{\gamma}$ is equivalent to solving a sparse coding problem for $\mathbf{X}$. Specifically, swe have*

$$\begin{aligned}&\min_{\boldsymbol{\gamma}\geq 0, \mathbf{X}} \frac{\|\mathbf{Y} - \mathbf{G}\mathbf{X}\|^2}{T\sigma_e^2} + \sum_{n=1}^{N}\sum_{t=1}^{T}\frac{X[n,t]^2}{T\gamma_n} + \mathbf{v}^T\boldsymbol{\gamma} + \rho\sum_{n=1}^{N}\gamma_n \\ &= \min_{\mathbf{X}} \frac{1}{T\sigma_e^2}\|\mathbf{Y} - \mathbf{G}\mathbf{X}\|^2 + 2\sum_{n=1}^{N}\sqrt{\frac{\rho + v_n}{T}}\|\mathbf{X}[n,:]\|\end{aligned}$$

Proof. The equivalence is obtained by substituting the optimal solution for $\boldsymbol{\gamma}$ derived in Eq. (10) into the original cost function Eq. (6). ∎

Based on this reformulation, the iterative algorithm proceeds as follows

Algorithm 1. *Initialization:* $\boldsymbol{\gamma}^{(0)} \in \mathbb{R}_+^N$.

- *Update* $\mathbf{v}$*:*

$$\mathbf{v}^{(i)} = \mathrm{Diag}\left[\mathbf{G}^T\mathbf{\Sigma}_y^{-1}(\boldsymbol{\gamma}^{(i)})\mathbf{G}\right]. \tag{11}$$

- *Minimize w.r.t* $\mathbf{X}$

$$\frac{1}{2\sigma_e^2}\|\mathbf{Y} - \mathbf{G}\mathbf{X}\|^2 + \sqrt{T}\sum_{n=1}^{N}\sqrt{\rho + v_n}\|\mathbf{X}[n,:]\|\ , \tag{12}$$

using K iterations of ISTA. Denotes the output by $\mathbf{X}^{(i)}$.

- *Update $\boldsymbol{\gamma}$:*

$$\gamma_n^{(i+1)} = \frac{\|\mathbf{X}^{(i)}[n,:]\|}{\sqrt{T(v_n^{(i)}+\rho)}}. \tag{13}$$

To analyze the convergence properties of Alg. 1, we introduce the following auxiliary objective function $\mathcal{F}$:

$$\mathcal{F}(\mathbf{X},\mathbf{v}) = \frac{\|\mathbf{Y}-\mathbf{G}\mathbf{X}\|^2}{T\sigma_e^2} + 2\sum_{n=1}^{N}\sqrt{\frac{\rho+v_n}{T}}\|\mathbf{X}[n,:]\| - w^*(\mathbf{v}). \tag{14}$$

with $w^*(\mathbf{v})$ given by Eq. (7). $\mathcal{F}$ is directly linked to the original SBL cost function $\mathcal{L}^{II}$, as stated in the following lemma.

Lemma 1. *If $(\mathbf{X},\mathbf{v})$ is a stationary point of $\mathcal{F}$, then $\boldsymbol{\gamma}$ defined as $\gamma_n = \frac{\|\mathbf{X}[n,:]\|}{\sqrt{(\rho+v_n)}}$ is a stationary point of $\mathcal{L}^{II}$.*

Proof. The equivalence follows directly from the relation:

$$\mathcal{F}(\mathbf{X}^{(i)},\mathbf{v}^{(i)}) = \mathcal{C}(\mathbf{X}^{(i)},\boldsymbol{\gamma}^{(i+1)},\mathbf{v}^{(i)})\ \forall i \geq 0.$$

Which is also true for a stationary point $(\mathbf{X},\mathbf{v})$ of $\mathcal{F}$ with $\gamma_n = \frac{\|\mathbf{X}[n,:]\|}{\sqrt{T(\rho+v_n)}}$ for all n. Hence, $(\mathbf{X},\boldsymbol{\gamma},\mathbf{v})$ is a stationary point of $\mathcal{C}$, and then $\boldsymbol{\gamma}$ satisfies the stationarity conditions of $\mathcal{L}^{II}$ as shown in [12]. ∎

Next, we show that $\mathcal{F}$ decreases monotonically during the iterations.

Lemma 2. *For all $i \geq 0$, we have*

$$\mathcal{F}(\mathbf{X}^{(i+1)},\mathbf{v}^{(i+1)}) \leq \mathcal{F}(\mathbf{X}^{(i)},\mathbf{v}^{(i)}) - \frac{L}{2}\|\mathbf{X}^{(i+1)}-\mathbf{X}^{(i)}\|^2.$$

Proof. Using the properties of ISTA, we have (see for example [9]):

$$\mathcal{F}(\mathbf{X}^{(i+1)},\mathbf{v}^{(i+1)}) \leq \mathcal{F}(\mathbf{X}^{(i)},\mathbf{v}^{(i+1)}) - \frac{L}{2}\|\mathbf{X}^{(i+1)}-\mathbf{X}^{(i)}\|^2. \tag{15}$$

Furthermore, since $\mathbf{v}^{(i+1)}$ minimizes $\mathcal{F}(\mathbf{X}^{(i+1)},\mathbf{v})$, the result follows. ∎

By combining the results above, we can state the following theorem on the convergence of the proposed algorithm.

Theorem 1. *Any accumulation point of the sequence $\{\boldsymbol{\gamma}^{(i)}\}$ is a stationary point of $\mathcal{L}^{II}$.*

Proof. By Lemma 2, the sequence $\{\mathcal{F}(\mathbf{X}^{(i)},\mathbf{v}^{(i)})\}$ is strictly decreasing. Moreover, since $\Sigma \geq \sigma_e^2 I$, the sequence $\{\mathbf{v}^{(i)}\}$ is necessarily bounded. Hence, $\mathcal{F}$ is lower bounded, ensuring the convergence to a local minimum F^*. Finally, the continuity of $\mathcal{F}$ ensures that any accumulation point satisfies the stationarity conditions of $\mathcal{L}^{II}$, as established in Lemma 1. ∎

In practice, monitoring the convergence of $\mathcal{L}^{II}$ can be numerically challenging due to the division by γ_n, which tends to zero for sparse solutions. As at each iteration $w^*(\mathbf{v}^{(i)})$ is given by:

$$w^*(\mathbf{v}^{(i)}) = \mathbf{v}^{(i)T}\boldsymbol{\gamma}^{(i-1)} - \log\left|\boldsymbol{\Sigma}_y(\boldsymbol{\gamma}^{(i-1)})\right|, \tag{16}$$

$\mathcal{F}$ can be stably monitored without divisions by γ_n.

To accelerate convergence, ISTA can be replaced by FISTA, which improves the convergence rate and reduces the overall computational cost.

B. Low SNR regime

The low SNR regime is particularly relevant in MEG applications, where the signal-to-noise ratio is often low due to measurement noise and the small amplitude of cortical activity. In this regime, as proposed in [12], the contribution of the data covariance to the log-determinant term in $\mathcal{L}^{II}$ becomes negligible. Specifically, we have:

$$\log|\boldsymbol{\Sigma}_y| = \mathrm{Tr}(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^T) + \mathcal{O}(SNR). \tag{17}$$

Substituting this approximation into the SBL objective, one can show that the resulting cost function $\mathcal{L}^{low}$ can be expressed as:

$$\mathcal{L}^{low}(\mathbf{X},\boldsymbol{\gamma}) = \frac{1}{T\sigma_e^2}\|\mathbf{Y}-\mathbf{G}\mathbf{X}\|^2 + \frac{1}{T}\sum_{n=1}^{N}\sum_{t=1}^{T}\frac{X[n,t]^2}{\gamma_n} + \sum_{n=1}^{N}(\|\mathbf{G}[:,n]\|^2+\rho)\gamma_n. \tag{18}$$

Minimizing $\mathcal{L}^{low}$ with respect to both $\mathbf{X}$ and $\boldsymbol{\gamma}$ yields the equivalent minimisation problem in $\mathbf{X}$:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y}-\mathbf{G}\mathbf{X}\|^2 + \sigma_e^2\sqrt{T}\sum_{n=1}^{N}\sqrt{\|\mathbf{G}[:,n]\|^2+\rho}\ \|X[n,:]\|\ , \tag{19}$$

which shows that in the low SNR regime, the problem reduces to a weighted ℓ_{21}-regularized least squares problem. The weights depend explicitly, and only, on the norm of the columns of $\mathbf{G}$ and the regularization parameter ρ.

IV. Numerical Results

In this section, we numerically compare the standard CHAMPAGNE algorithm against Alg. 1. We also compare two different implementations of the ℓ_{21}-solver in Alg. 1, a standard FISTA solver and celer [13], a highly optimized solver leveraging a duality-based approach. It should be noted that celer does not provide a solver for reweighted ℓ_{21}, which causes our setup to introduce a variable change for that implementation. This may impact speed and result quality for celer when $T > 1$.

The experiments are run on an Ubuntu system equipped with an i7-10750H processor and 16 GB of RAM. All benchmarks are managed using benchopt [14]. The source code to reproduce our results can be found at https://gitlab.inria.fr/dsechet/sbl_as_sparse_coding.

A. Compressed sensing

We first compare the algorithms on a simple compressed sensing scenario, with $\mathbf{G}$ a 300×1000 column-normalized standard Gaussian random matrix, 90% sparsity, and $T = 1$. Signals are generated with low noise (average SNR of 25 dB in sensor space). All three implementations show their best

performance for $\rho = 0$. Performance is aggregated across 1000 random samples: for each experiment, both $\mathbf{G}$ and $\mathbf{X}$ are redrawn. Performance over time is shown in Fig. 1. Our algorithm is significantly faster than CHAMPAGNE and converges to a slightly better solution in terms of SNR.

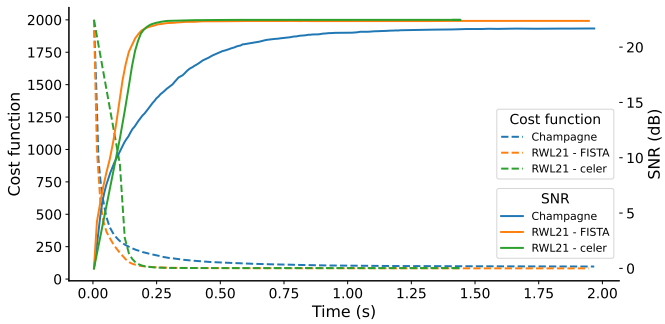

Fig. 1. Median SNR and $\mathcal{F}$ evolution across 1000 compressed sensing problems for $T = 1$ and $\rho = 0$

B. MEG setting

To evaluate the robustness of our algorithm, we tested it in a realistic MEG inverse problem scenario. MEG inverse problems usually involve highly sparse solutions in space but are particularly challenging due to their high noise levels and ill-conditioned sensing matrices. We use a column-normalized matrix $\mathbf{G}$ of size 305×1884, derived from the sample subject in the MNE-python dataset [15], and generate a sparse signal in space, with 0.5% of active sources. The active sources are populated for $T = 10$ timesteps using a randomized frequency, amplitude, and phase sinusoid. The noise variance is chosen so the data's SNR is close to 0.3 dB.

Given the high noise level, we compare the standard SBL algorithm to its low-SNR approximation, where we treat σ_e as a hyperparameter σ_0, effectively eliminating the need for ρ, which we set to zero. In this regime, the approximate algorithm tends to induce excessive sparsity, often leading to trivial solutions. While adjusting ρ could mitigate this, our experiments indicate that the optimal weight would typically be negative. Performance is aggregated across 30 trials, with $\mathbf{G}$ fixed while $\mathbf{X}$ and the noise are resampled.

In Fig. 2, we plot the time to convergence of various algorithms, along with the SNR at convergence. Convergence is monitored using the cost function introduced in Eq. (19). The results highlight a tradeoff between speed and solution quality. While the standard SBL formulations achieve much better solution quality, they converge more slowly than the low-SNR approximation methods.

Celer achieves the fastest convergence in low-SNR mode and also achieves slightly better results than CHAMPAGNE. That tendency is inverted for the standard SBL formulation, with CHAMPAGNE slightly outperforming celer while being faster. FISTA performs poorly on this problem, converging much slower than the other two implementations, both in SBL and low-SNR mode. Our choice of using σ_0 as a hyperparameter instead of directly setting $\sigma_0 = \sigma_e$ in the low-SNR approximation is vindicated, as we can improve the solutions' quality without any significant impact on runtime with an optimal choice of σ_0. In Fig. 3, we can see how the SNR evolves over time for both the standard SBL algorithm and the low-SNR approximation with an optimal σ_0.

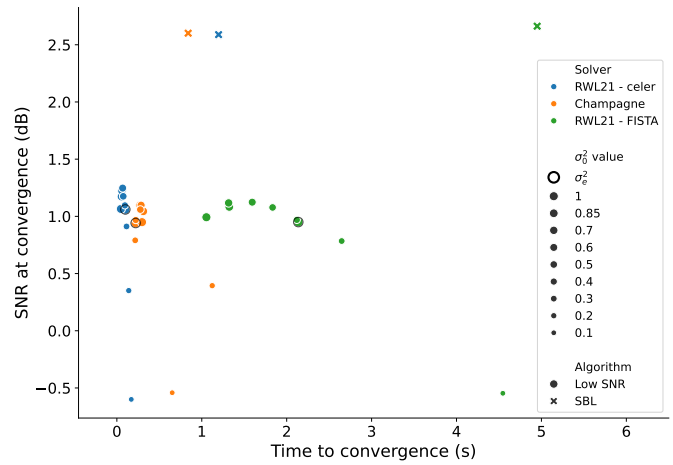

Fig. 2. Median SNR and time to convergence accross 30 MEG problems

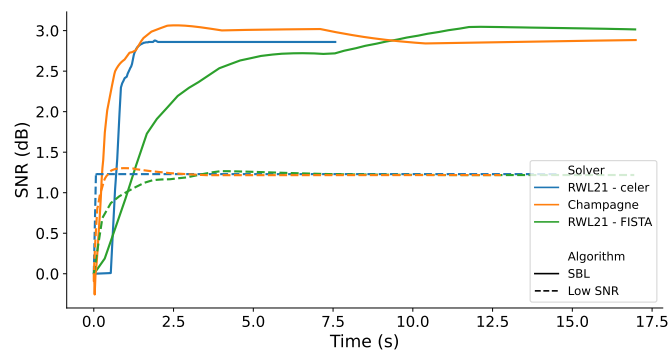

Fig. 3. Median SNR evolution across 30 MEG problems. For the low-SNR regime, the best σ_0 is chosen. It happens to be $\sigma_0^2 = 0.6$ for all solvers.

V. Conclusion

We introduced a reweighted sparse coding formulation of the SBL optimization problem, leading to an efficient iterative algorithm. Our results show that for $T = 1$, the proposed approach significantly outperforms CHAMPAGNE regarding computational efficiency while ensuring exact sparsity. For $T > 1$, performance depends on the choice of the iterative thresholding solver: while FISTA proves suboptimal, using Celer yields competitive results. In the low-SNR regime, the non-reweighted formulation consistently achieves superior efficiency. An unexpected observation is that setting $\rho = 0$ systematically leads to the best results, which warrants further investigation. Future work will also focus on analyzing the convergence properties of the iterates by leveraging the Kurdyka-Łojasiewicz framework.

REFERENCES

[1] S. Baillet, "Magnetoencephalography for brain electrophysiology and imaging," *Nature neuroscience*, vol. 20, no. 3, pp. 327–339, 2017.

[2] M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting magnetic fields of the brain: minimum norm estimates," *Medical & biological engineering & computing*, vol. 32, pp. 35–42, 1994.

[3] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[4] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1. IEEE, 1994, pp. 41–44.

[5] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 68, no. 1, pp. 49–67, 2006.

[6] W. Ou, M. S. Hämäläinen, and P. Golland, "A distributed spatio-temporal eeg/meg inverse solver," *NeuroImage*, vol. 44, no. 3, pp. 932–946, 2009.

[7] A. Gramfort, M. Kowalski, and M. Hämäläinen, "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods," *Physics in Medicine & Biology*, vol. 57, no. 7, p. 1937, 2012.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[9] P. Tseng, "Approximation accuracy, gradient methods, and error bound for structured convex optimization," *Mathematical Programming*, vol. 125, no. 2, pp. 263–295, 2010.

[10] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[11] J. Owen, H. Attias, K. Sekihara, S. Nagarajan, and D. Wipf, "Estimating the location and orientation of complex, correlated neural activity using MEG," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 21, 2008.

[12] A. Hashemi, C. Cai, G. Kutyniok, K.-R. Müller, S. S. Nagarajan, and S. Haufe, "Unification of sparse Bayesian learning algorithms for electromagnetic brain imaging with the majorization minimization framework," *NeuroImage*, vol. 239, p. 118309, 2021.

[13] M. Massias, A. Gramfort, and J. Salmon, "Celer: a fast solver for the lasso with dual extrapolation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 3321–3330.

[14] T. Moreau, M. Massias, A. Gramfort, P. Ablin, P.-A. Bannier, B. Charlier, M. Dagréou, T. Dupré la Tour, G. Durif, C. F. Dantas, Q. Klopfenstein, J. Larsson, E. Lai, T. Lefort, B. Malézieux, B. Moufad, B. T. Nguyen, A. Rakotomamonjy, Z. Ramzi, J. Salmon, and S. Vaiter, "Benchopt: Reproducible, efficient and collaborative optimization benchmarks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[15] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013.