Comparison and analysis of optimization techniques to induce structured sparsity in source localization for M/EEG data

Pavan S Holur, ECE 236C

May 2019

1 Introduction

This project investigates structured sparsity in the inverse problem of source localization, provided noisy and under-determined M/EEG outputs from non-invasive sensors placed on the head. The data is under-determined due to the number of modelled sources in the brain outnumbering the number of viable sensors on the scalp. Further, the project compares and contrasts different optimization techniques in solving the resulting optimization problem for this localization, and also analyzes associated techniques such as re-weighted regularization, active set heuristics and the response to regularization with respect to number of time samples in the optimization.

2 Background

The inverse problem of accurately detecting the activity of neural currents in the brain with the sequential response rendered by a non-invasive sensor network, is an area useful in establishing cause to many psychological issues including depression, anxiety, and hyper-activity [5]. Unfortunately however, the *post* data observed by the sensor grid describes a highly under-determined and noisy set of equations to solve for brain activity; i.e the problem is ill-posed.

Despite this limitation, recent advances in optimization methods [3] and apriori knowledge of sparsity in the discrete set of possible neural current locations in the brain (modelled as electric dipoles) [7,8] help better pose the problem of mapping. Along with sparsity, which improves the results of an under-determined set of equations in space, we need to establish temporal reliability as well, to mitigate the effects of additive noise, which yield different sparse solutions across successive time stamps in a window of observation. In other words, we need to impose an implicit constraint on spatial sparsity by looking at the trends over time.

3 Problem Formulations

3.1 The First Attempt: Lasso-Regularized Least Squares

As a primer to develop few of the convergence algorithms required in the successive structured sparsity formulations, we first consider the L1-norm constraint along with a simple 2-norm problem to solve for a sparse inverse solution to a given output. Sparsity in convex optimization problems was first introduced in [1]. Simply the objective is given below with the minimization given in equation 2 as a sum of two functions $f + g$. It is important to note the lasso criterion is convex, but not strictly convex if $A^T A$ is not full rank; thus the lasso solution may not be unique. Further, the stopping criterion for these algorithms includes the 2-norm of the primal residual as suggested in [4].

$$\mathbf{y} = A\mathbf{x} + \mathbf{e} \tag{1}$$

$$\underset{\mathbf{x}\in S}{\text{minimize}}\ \frac{1}{2}||A\mathbf{x} - \mathbf{y}||_2^2 + \gamma||\mathbf{x}||_1 \tag{2}$$

It is well understood that the proximal operator of the L1 norm is separable and derived from the soft-max function given by equation 3. Furthermore, the least squared error (LSE) function is differentiable and convex.

$$\text{Prox}_{\gamma\|\cdot\|_1}(\mathbf{x})_i = \text{sign}(x_i)\max(|x_i| - \gamma, 0) \tag{3}$$

The simple proximal operator coupled with LSE, makes (Accelerated) Proximal Gradient Descent ((A)PGD) and Alternating Directed Method of Multipliers (ADMM) suitable algorithms to consider to solve this optimization problem. Extensive information on these methods and their proofs of convergence can be found in [9].

3.1.1 Implementation of (Accelerated) Proximal Gradient Descent

For PGD, the main iterative step is characterized by equation 4. It is important to understand that PGD merely approximates the optimal value of x for the L1-norm and the quadratic Taylor approximation of the LSE term. For this implementation, we created a backtracking line search to find the optimal γ^k. The verification for bounds to accept a parameter is given in equation 15. The G function is simply the difference of existing x with a x' solved with the new γ parameter.

$$\mathbf{x}^{k+1} = \text{prox}_{\gamma g}[\mathbf{x}^k - \gamma^k \nabla f(\mathbf{x}^k)] \tag{4}$$

$$f(\mathbf{x} - \gamma G_\gamma(\mathbf{x})) > f(x) - \gamma\nabla f(\mathbf{x})^T G_\gamma(\mathbf{x}) + \frac{\gamma}{2}||(G_\gamma(\mathbf{x})||_2^2 \tag{5}$$

It is known the objective is convex, with the function f differentiable and L-smooth and g provably convex and closed [5]. As a result the above methods are guaranteed to converge. Also acceleration can be introduced by the following extension of x. The constant ζ is made to be roughly dependent on k, reducing the acceleration for further iterates.

$$\hat{x}^{k+1} = \mathbf{x}^{k+1} + \zeta(\mathbf{x}^{k+1} - \mathbf{x}^k) \tag{6}$$

It is expected that ζ acts like a "stability" parameter; this property is further analyzed later. Also the complexity improvements observed in the accelerated method are also analyzed.

3.1.2 Implementation of Alternating Direction Method of Multipliers

As an alternating system of projections with the augmented lagrangian functions, the ADMM problem is reformulated as in equation 7.

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & \frac{1}{2}||A\mathbf{x} - \mathbf{y}||_2^2 + \gamma||\mathbf{z}||_1 \\ \text{subject to} \quad & \mathbf{x} - \mathbf{z} = 0 \end{aligned} \tag{7}$$

To solve this problem, it is well documented that the following reductions satisfy the optimization problem. Also note in general, the update for block of variables reduces to the prox update whenever the corresponding linear transformation is the identity matrix. This is thus equivalent to the Douglas-Rachford method.

$$\begin{aligned} \mathbf{x}^{k+1} &= (A^T A + wI)^{-1}(A^T\mathbf{y} + w\mathbf{z}^k - \mathbf{t}^k) \\ \mathbf{z}^{k+1} &= \text{prox}_{g,\frac{\gamma}{w}}(\mathbf{x}^{k+1} + \mathbf{t}^k/w) \\ \mathbf{t}^{k+1} &= \mathbf{t}^k + w(\mathbf{x}^{k+1} - \mathbf{z}^{k+1}) \end{aligned} \tag{8}$$

The proximal operator on g has been solved in the PGD section; as a result, we simply write a function for x^{k+1} and find a suitable w to finish the ADMM implementation. The complexity and rate of convergence is analyzed in later sections.

3.2 Structured Sparsity using L-21 Norm Regularization

Expanding on the formulation in the previous section, we construct the optimization problem related to structured source estimation given under-determined and noisy sensor data. The linear equations that yield the post data observed by sensors (M) is given by equation 9. Note that M is the observed matrix across sensors and time, G is the corresponding well-defined gain matrix, and the additive noise E is not well modelled. This formulation is further described in [5].

$$M = GX + E \tag{9}$$

In general, if X is of dimension [m x n] where m is the number of sources and n is the number of time samples, then G is the matrix that maps sources to sensors. In other words, G would then be wide as there are more sources than sensors. Following this line of argument, if G is of dimension [k x m], the error matrix is of dimension [k x n]. Despite G being wide, we establish convergence to a unique solution by a regularizer that models this ill-posed problem better with the L-21 norm that makes the objective strictly convex [5].

To yield the sparse and reasonably time-invariant detection of brain activity, the formulated optimization problem is provided below, along with the suggested weighted regularization metric $L_{\mathbf{w},21}$, that fuses the L2-norm for time robustness and L1-norm for spatial sparsity. The weights w are found per source channel. This variable represents the depth bias for different sensors on the head.

$$X^* = \underset{X}{\operatorname{argmin}} \frac{1}{2}||M - GX||_F^2 + \lambda||X||_{\mathbf{w},21} \tag{10}$$

$$||X||_{\mathbf{w},21} = \sum_s \sqrt{\sum_t w_s X_{s,t}^2} \tag{11}$$

The proximal operator applied to this non-differentiable regularizer is simple enough to compute with the following resulting equation derived in [5]. The separability and low cost of the proximal operator of this cascaded regularizer, allows sophisticated optimization solvers to solve equation 10 at low complexity.

$$X = \mathbf{prox}_{||\cdot||_{\mathbf{w},21}} Y \iff x_{s,t} = y_{s,t}(1 - \frac{\lambda\sqrt{w_s}}{||\mathbf{y}_s||_2})^+ \tag{12}$$

3.2.1 Additions to the proposed convergence algorithms in Section 3.1

Instead of equations 4 and for the acceleration in equation 6, the following expressions are suitable for matrix functions. Note that these are just compact ways of writing a vectorized form of the matrix objective, since the objective functions are separable. Similarly for the ADMM, the problem is formulated with matrix variables. More analysis is provided in [10].

$$\begin{aligned} X^{k+1} &= \text{prox}_{\gamma g}[X^k - \gamma^k \nabla f(X^k)] \\ \hat{X}^{k+1} &= X^{k+1} + \zeta(X^{k+1} - X^k) \end{aligned} \tag{13}$$

Note that because backtracking is used, the algorithm empirically converges. For a fixed step size however, one may consider the following parameter.

$$\frac{1}{\gamma^k} = \max \lambda_{max}(G^T G) \triangleq \text{Lip} \tag{14}$$

Lastly, in the backtracking bound described in 5, the matrix form includes the following edit to account for the matrix variables:

$$f(X - \gamma G_\gamma(X)) > f(X) - \gamma\langle \nabla f(X), G_\gamma(X)\rangle + \frac{\gamma}{2}||(G_\gamma(X)||_F^2 \tag{15}$$

3.2.2 Other Implementations

We includes re-weighting the weights w of the L-21 norm per iteration depending on the online viability of a specific sensor to perhaps have an impact in the source localization. The re-weighting is done on only 1 of the algorithms (in our case PGD) but it can work on any of the methods. The re-weighting is given by equation 16. The ϵ stabilizes the algorithm preventing the weights from stretching to ∞. The convergence of this method is given in [2].

$$w_i^{k+1} = \frac{1}{2||X_i^k||_2 + \epsilon} \; \forall i \in S \tag{16}$$

While manually implementing PGD, APGD and ADMM, we also consider AS-APGD [5], the state-of-the-art algorithm used on M/EEG data. The benefit of AS-APGD includes the assumption of an active set (AS), or a limited number of sources assumed to be contributing to the response observed. This is more memory efficient than ADMM approaches as the memory required for compute relies on the gram matrix G^TG which is quadratically larger in the later due to the presence of more sources. As shown in [5], the AS-APGD algorithm works as follows.

1. Pick τ number of random sources to consider as active sources. This would violate the dual feasibility of the original formulation if the selection is wrong.

2. Set all other source contributors in X to 0. Then if the measured duality gap is above allowed threshold there are few X components missing.

3. Find a group of elements in X that worst violate the following Primal-Dual Constraint and add these sources to the active set. This is in fact dual feasibility.

$$||\text{diag}(\mathbf{w})^{-1}G^T(M - GX)||_{2,\infty} \leq \lambda \tag{17}$$

4. Repeat until KKT conditions are met. This is an optimal solution of the full problem. Further, this is bound to converge as the largest set of X includes all sources, with which we assume the optimization is well-behaved.

3.3 Few Associated Proofs

3.3.1 Deriving the proximal operator for the L-21 norm

We know that the following holds by definition of the proximal operator. Assume that $\mathbf{x}$ is the row-wise vectorized form of X to apply to L-21 norm. We also know that the L-21 norm is convex due to the homogeneity of the norm. Description of this implementation of the L-21 norm is provided in equation 11. Every i refers to 1 source in the set of sources.

$$\begin{aligned} \text{prox}_{\lambda||.||_{\mathbf{w},21}}(\mathbf{y}) &= \underset{\mathbf{x}}{\text{argmin}}(||\mathbf{x}||_{\mathbf{w},21} + \frac{1}{2}||\mathbf{y} - \mathbf{x}||_2^2) \\ &= \underset{\mathbf{x}_i.i\in[0,|S|]}{\text{argmin}}\,(||\mathbf{x}_i||_{\mathbf{w},2} + \frac{1}{2}||\mathbf{y}_i - \mathbf{x}_i||_2^2) \end{aligned} \tag{18}$$

For every x_{ij}, there exists only 1 row of the matrix X relevant in the computation of the L-21 norm. Within this row, the weight w_i is a constant. It is also worth noting that the above reduction is allowed due to separability of x, because minimizing each block vector is equivalent to minimizing the global objective including the 1-norm. The final result reduces to the 2-norm problem. Using the Moreau decomposition and the self-conjugating 2-norm ball, we can directly write,

$$X = \mathbf{prox}_{||.||_{\mathbf{w},21}}Y \iff x_{s,t} = y_{s,t}(1 - \frac{\lambda\sqrt{w_s}}{||\mathbf{y}_s||_2})^+ \tag{19}$$

3.3.2 Deriving the dual of the primal objective and the feasibility constraint for AS-APGD

Firstly, to solve for the dual problem, it is useful to solve for the dual norm of the L-21 norm regularizer, to use existing results on simple Lasso-regularization and then form the dual. Therefore, with a similar row-wise vector of X and $i \in |S|$,

$$\begin{gathered} ||\mathbf{y}||^*_{\mathbf{w},21} = \max_{||\mathbf{x}||_{\mathbf{w},21}=1} (\mathbf{x}^T\mathbf{y}) = \max_{\sum \sqrt{w_i} a_i = 1} \sum \mathbf{y}_i^T \mathbf{x}_i \\ \text{where } a_i = ||\mathbf{x}_i||_2, a_i \geq 0 \end{gathered} \tag{20}$$

This can further be reduced to get the $[2, \infty]$ norm as shown in the steps below using Cauchy-Schwartz inequalities twice - one for 2-norm dual and one for the $1 - \infty$ norm dual.

$$\begin{gathered} \max_{\sum \sqrt{w_i} a_i = 1} \sum \mathbf{y}_i^T \mathbf{x}_i = \max_{\sum \sqrt{w_i} a_i = 1} \sum ||\mathbf{y}_i||_2 a_i \\ \max_{\sum \sqrt{w_i} a_i = 1} \sum ||\mathbf{y}_i||_2 a_i \sqrt{w_i}/\sqrt{w_i} = \max_{\sum \sqrt{w_i} a_i = 1} \sum ||\mathbf{y}_i/w_i||_2 a_i \sqrt{w_i} \\ \text{let W} = \mathbf{diag}(\text{w}) \end{gathered} \tag{21}$$

Then, we can make the following substitutions to observe the dual norm of a 1-norm which is in fact the ∞ norm. If,

$$\begin{gathered} \mathbf{z} = \begin{bmatrix} . \\ . \\ ||\mathbf{y}_i/w_i||_2 \\ . \\ . \end{bmatrix}, \mathbf{q} = \begin{bmatrix} . \\ . \\ a_i\sqrt{w_i} \\ . \\ . \end{bmatrix} \\ \max_{||\mathbf{q}||_1 = 1} \mathbf{z}^T\mathbf{q} = ||\mathbf{z}||_\infty \end{gathered} \tag{22}$$

If it is mutually agreed that given a mixed norm of the following form: [a,b], 'a' applies on the rows and the norm 'b' applies on the resulting column vector, the following equivalence holds.

$$||Y||^*_{\mathbf{w},21} = \begin{bmatrix} . \\ . \\ ||\mathbf{y}_i/w_i||_2 \\ . \\ . \end{bmatrix}_\infty = ||W^{-1}Y||_{2,\infty} \tag{23}$$

Knowing that the conjugate of a norm is the indicator of the dual norm ball, we can now derive the dual problem to the provided objective in equation 10 similar to the simple lasso-regularizer.

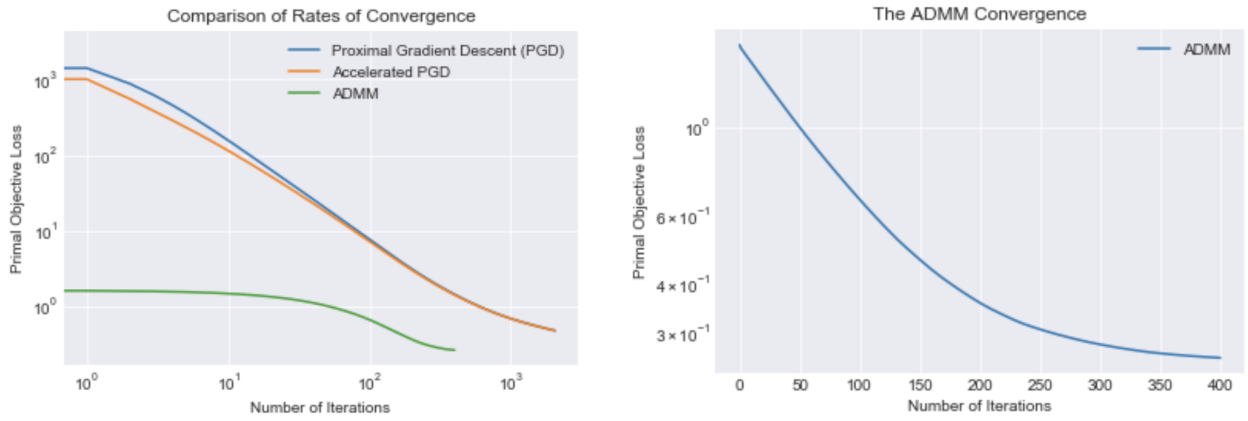
$$\begin{aligned} P^* &= \min_X \frac{1}{2}||M - GX||_F^2 + \lambda||X||_{\mathbf{w},21} \\ Q^* &= \min_H ||M - H||_F^2 \qquad\qquad [\text{subject to } ||W^{-1}G^TH||_{2,\infty} \leq \lambda] \end{aligned} \tag{24}$$

Solving for the dual problem, we find the required constraint in equation 10 by computing the KKT Lagrangian sub-differential function, done in [12] and substituting for H.

4 Analysis and Results

4.1 Experiments on Lasso Regularized Least Squares

This analysis and result is applicable for a single time stamp of M/EEG data which is sparse and the goal is once again the inverse problem of trying to find the sparse solution of the given output vector. These solutions are found for a [1000 x 1000] dimension invertible gain matrix A. As covered in section 3.1, the comparative rates of convergence as a result of PGD, APGD and ADMM are shown in a comparative plot in figure 1.a. The algorithms' functionality was verified by the convergence graphs and the matching of modelled versus retrieved source variables. Note that the acceleration does improve the convergence. Theoretically, $O_{PGD}(k) = 1/k$ and $O_{APGD}(k) = 1/k^2$. Here k is the iteration number. This implies that the acceleration can help the problem converge faster.



(a) ADMM finds a different loss compared to the other two methods and this is to be expected.

(b) The ADMM solution is the most precise solution and reaches an acceptable loss before max iterations are reached.

Figure 1: Convergence Rates for the Lasso-Regularized Least Squares Objective

Furthermore one can observe the unique property of APGD that creates the extrapolating effect on the solution variable, which in turn results in oscillating objectives for aggressive accelerations. These oscillations may sometimes empirically fail to match the complexity mentioned earlier shown in 3.

It is also key to observe that the above plots are drawn with a very low relative tolerance to successive iterations as the bounding threshold (RELTOL = 0.0001), with maximum iterations set to 10000. It is notable that duality gap is the best delimiter to stop iterations rather than gradient thresholds as we are using Lasso regularizers which are non-differentiable. As a result, the PGD and APDG reach a moderate optimal and time-out whereas the ADMM implementation manages to find a high-precision result in roughly 400 iterations, much earlier than the PGD and APGD algorithms.

Indeed, ADMM optimizers yield moderate precision in excellent time complexity but high precision in poor time complexity [4]. Under optimal constraints and moderate expectations, ADMM convergences have shown to be roughly linear or sub-linear. The precision

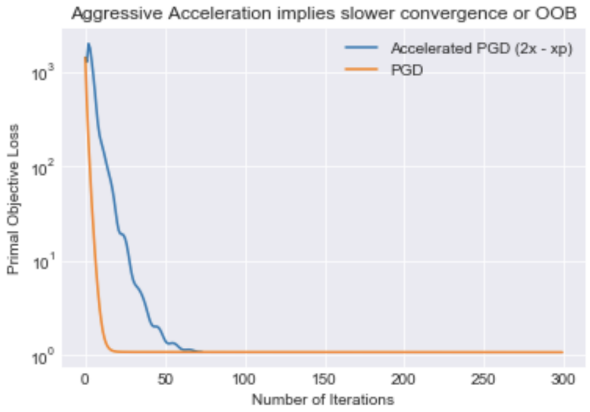


Figure 2: Challenges in setting an acceleration constant in FISTA

is, however, vital to our studies on M/EEG data, and PGD, APGD or any other model may converge to a high-precision result faster as seen in figure 4.a.

4.2 Structured Sparsity on Artificially-Modelled M/EEG Data

To artificially model the M/EEG data, a forward problem is first considered that involves setting up a sparse vector and horizontally stacking the vector to create a structured and sparse matrix (see figure 4.b). The dimensionality and enhancements to this matrix problem are discussed further in section 3.2. We use dimension of G as [10 x 30] with the number of time stamps equalling 50 samples. This is roughly 2x the factor of source to sensor to time ratio utilized in general M/EEG applications. The modelled error is a matrix Gaussian noise set at 10dB signal-to-noise ratio (SNR).

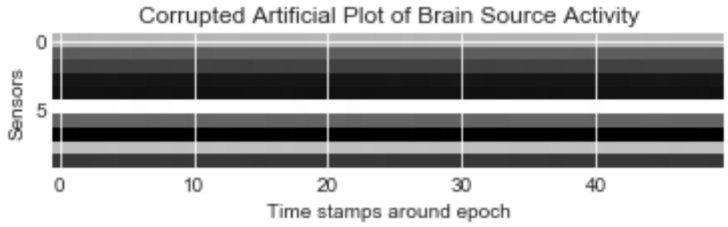
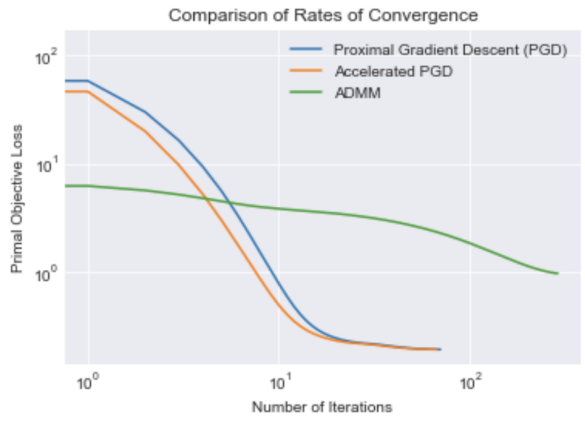
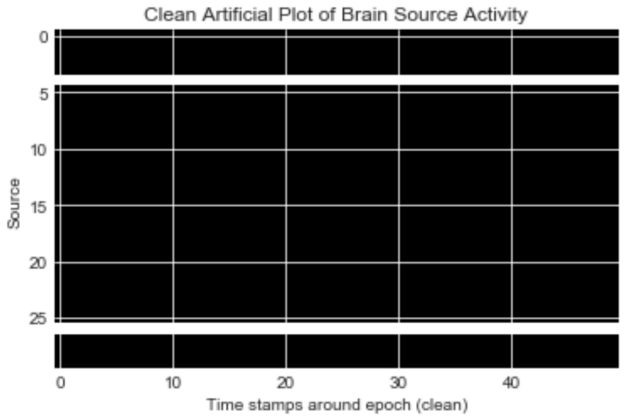


Figure 3: The output M that is used to retrieve the source, knowing the gain matrix G and the property that the source is structured sparse

Once again, the 3 optimization methods are considered. In particular, this comparison does not utilize re-weighting mentioned in equation 16 or solve for an approximation of the initial weights for the regularizer using noise co-variance estimation [5]. A hard thresholding function has been applied to the output to generate provably similar results to 4.b. This may not be an optimal primal solution but shows proof of concept. Also the maximum iterations is 500, with the same relative tolerance as before. The convergence plot is provided in figure 4.a.



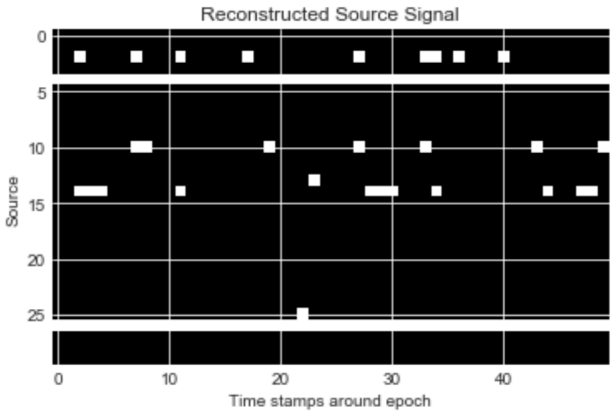
(a) The convergence comparisons for matrix LSE and L-21 mixed norm regularization



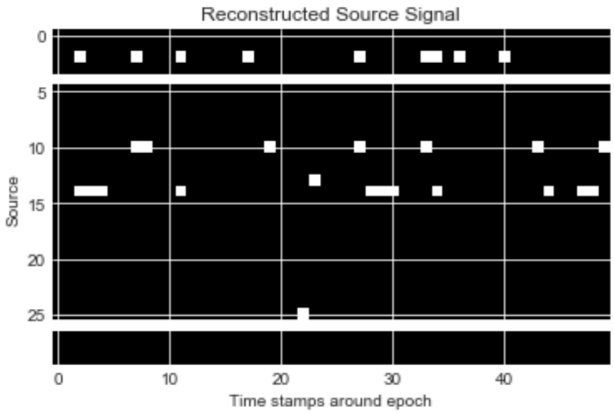
(b) The modelled source signal: Roughly, 1 in 70 sources are active for a given task [5]

Figure 4: Convergence Rates for the L-21 norm regularized matrix Least Squares Objective and the original source signal X

It should be noted that in figure 4.a, the ADMM once again results in a "moderate" solution within 10 iterations but takes too many more iterations to reach a "high precision" solution. This lack of fidelity is not suitable to us, and at this point we are content with A/PGD methods that take moderate time to converge to an excellent solution. The inverse results are provided below for verification. Note that the ADMM does not have a clean result.



(a) Inverse solution for PGD after hard-thresholding



(b) Inverse solution for APGD after hard-thresholding

Figure 5: The inverse solutions verify that the L-21 regularizer creates structured and sparse solutions for ISTA-based algorithms (hard thresholded for visual effect)

The results of ADMM are given below. Note that the ADMM implementation also requires the inversion of a large $A^T A$ matrix as shown in equation 8. This creates large complexity per iteration as the gram matrix has the dimension of the number of sources, which is often too large. Many quadratic programming approaches exist today to solve these

inverses effectively, but we do not study them in this context, frankly because the number of iterations are too large for our required fidelity.

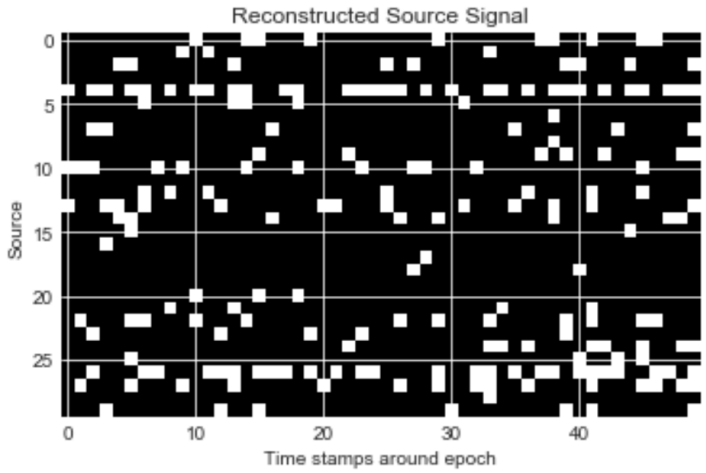


Figure 6: The ADMM inverse source modelled after the algorithm times out at 100 iterations. The sparse solution formation is barely noticeable, and the algorithm is both slow per iteration and takes too many iterations for a high fidelity source modelling.

4.2.1 Iterative Re-weighting on the L-21 norm

Lastly, we have a look at the iterative re-weighting of the L-21 norm as described in equation 16. We hope to find more efficient (in iterations) optimization using this method as discussed in [5]. It is well established that re-weighting was initially described to solve iterative convex surrogate optimization problems in lieu of non-convex optimizations, with the weights ensuring the convexity of the regularizer (i.e for norms between 0 and 1). However, the L-21 norm is provably convex.

Below in figure 7 is the comparative rates of convergence of PGD and re-weighted PGD for the same problem as above. The initial weights are unity; however in general purpose solvers, this initialization is done by hot-spot heuristics [11].

4.3 Structured Sparsity on Real M/EEG Data

Similar to the above approach, we now consider real M/EEG data sets obtained through an open source software MNE [6] (minimum norm estimator), that allows trimming and manipulating raw sensor data, crucial to creating reliable data sets. A general concern is understanding how exactly 3D sources in the brain can be mapped to a vector of sources. After all, the sensors are mapping the sources so that the inverse problem is a one to many mapping. Recent advances in neuroscience have since mitigated this issue, providing discriminant features (including depth) to model vectorized 3D mappings to the brain. For more information on this map, the reader can consult papers by Ou et al. in [8]. Firstly, the M/EEG gain matrices (which are provided with datasets) are concatenated to form a larger gain matrix. The gain matrices are provided in figure 8.

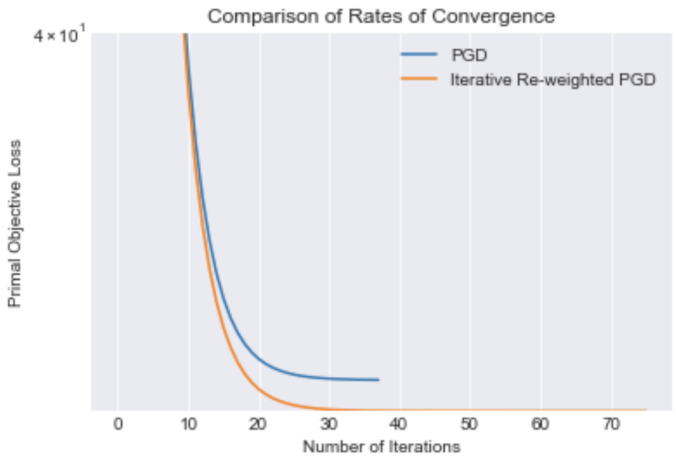


Figure 7: Notice that the weights are high for those sources tempted to be sparse. This further induces sparsity in those sources. The "feedback" helps in quicker and more efficient convergence.

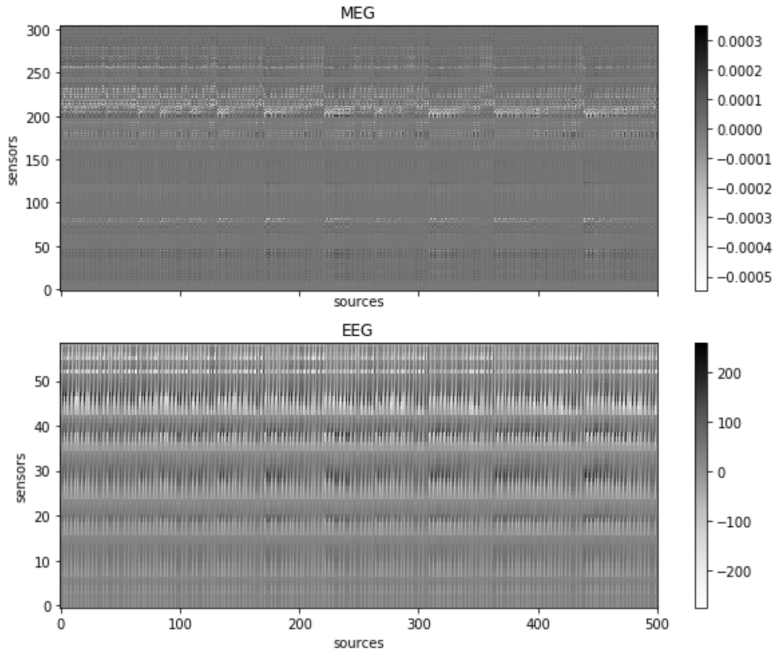


Figure 8: These noisy gain matrices can indicate why inverse problems are difficult to solve. [370 x 8000]

Covariance methods as explained in [5] can help find the hot-spot weights for the initialization of the sensor weights in iterative re-weighting implemented in section 4.2.1. Other details about the implementation include:

- M/EEG data is collected for left-visual stimulus to the subject.
- For every epoch of stimulus we collect 500ms of time stamps, from -100ms to 400ms

with 0ms being the instant of stimulus.

- Lastly, we consider APGD versus AS-APGD; the later describes a heuristic to consider more sparse vectors in X incrementally to satisfy KKT conditions.

4.3.1 Few Interesting Results

The first comparison is between applying and avoiding the heuristic for the active set in APGD. The general heuristic considered was 10; i.e $\tau = 10$. Source localization was as predicted faster with AS-APGD, but converged to the same result. In our case, time of convergence can be used as a suitable metric as number of iterations is not really comparable.

Optimization Algorithm	Time to Converge (s)
APGD	7.04
AS-APGD	1.93

Table 1: The AS-APGD is nearly 4x faster with the same results as APGD

Similarly, to examine the exact effect of the active set, we plot the time taken to converge versus the size of the active set, and the relationship can be observed in the graph 9.

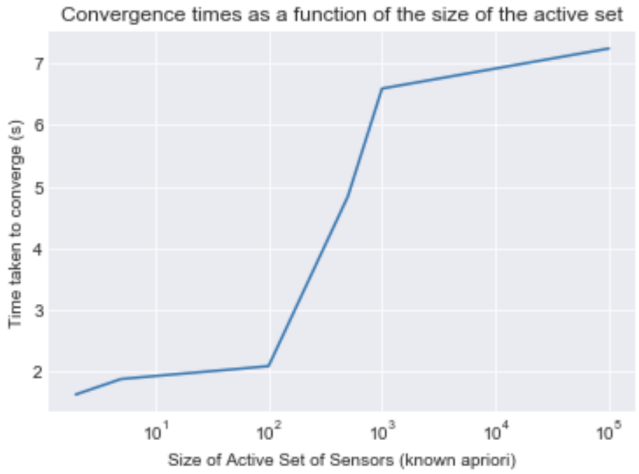


Figure 9: The rate of convergence worsens as the a-priori size of the active set worsens

As shown in figure 10 the correct active set is found quite abruptly after a certain number of time samples are captured around each epoch. This is when the regularizer "kicks" in and the structured sparsity is emphasized in the active set. The active set increases initially, due to the poor contributions of the regularizer; the time samples create more noise than structure until the regularizer starts to contribute. Lastly, an ideal source localized inverse solution is imaged in figure 11. Note that the re-weighting improved the source localization. With AS-APGD, re-weighting halved the number of active sources from 4 to 2 on left visual stimulus to a subject.

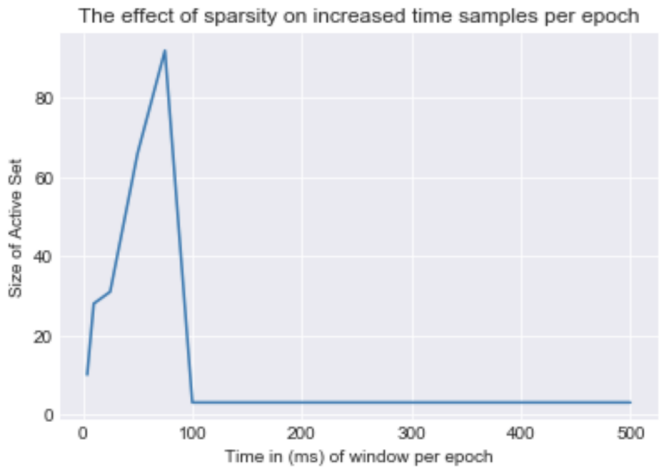


Figure 10: The variation of the optimal active set provided varied time windows

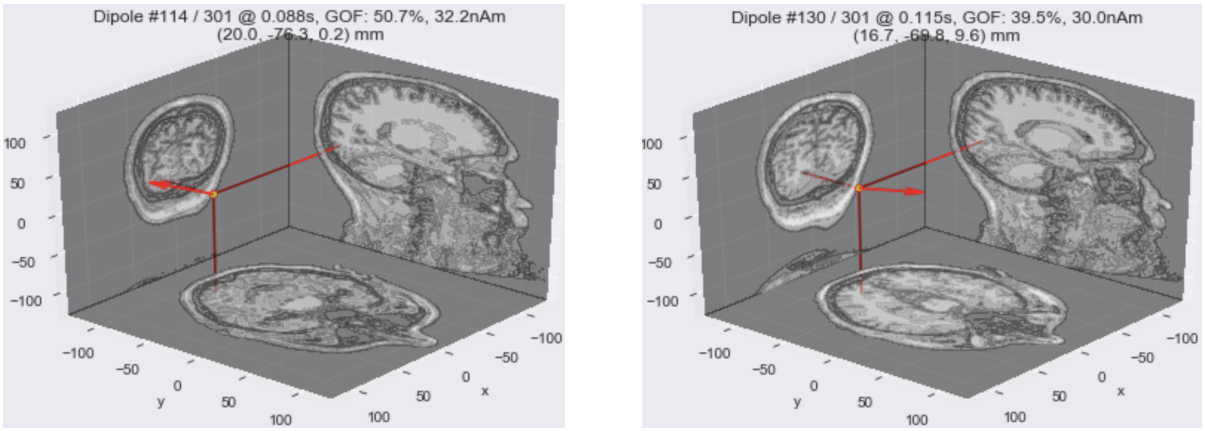


Figure 11: The 2 sources found by solving the inverse problem and projected onto a brain map using neuro-scientific transformations

5 Conclusions

Recent advances in optimization techniques and neuroscience have made source localization by solving the inverse problem a precise technology. Despite this, online localization is less understood. While PGD, APGD and AS-APGD can resolve sources within 5 seconds of solver techniques using concepts like structured sparsity, active set iteration and iterative re-weighting, this timing is exclusive of finding and isolating epochs, estimating or knowing the gain matrix, and solving a source weighting pre-instance. Moreover in online applications, because the orientations of the sensors on the scalp can deviate quite a bit, the gain matrices can vary and it is difficult to model these matrices online. Lastly, ADMM implementations, though often fast seem to be not precise enough in low time complexity for this application. Furthermore, inverting a large gram matrix is difficult, making this base ADMM implementation futile.

References

[1] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5:19–53, 2011.

[2] A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[5] Alexandre Gramfort, Matthieu Kowalski, and Matti Hämäläinen. Mixed-norm estimates for the m/eeg inverse problem using accelerated gradient methods. *Physics in Medicine and Biology*, 57(7):1937–1961, 2012.

[6] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hämäläinen. Mne software for processing meg and eeg data. *NeuroImage*, 86:446 – 460, 2014.

[7] Stefan Haufe, Vadim V. Nikulin, Andreas Ziehe, Klaus-Robert Müller, and Guido Nolte. Combining sparsity and rotational invariance in eeg/meg source reconstruction. *NeuroImage*, 42(2):726–738, 2008.

[8] Wanmei Ou, Matti S Hämäläinen, and Polina Golland. A distributed spatio-temporal eeg/meg inverse solver, Feb 2009.

[9] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[10] Jian Pu, Jun Wang, Yu-Gang Jiang, and Xiangyang Xue. Multiple task learning with flexible structure regularization. *Neurocomputing*, 177:242 – 256, 2016.

[11] Daniel Strohmeier, Yousra Bekhti, Jens Haueisen, and Alexandre Gramfort. The iterative reweighted mixed-norm estimate for spatio-temporal meg/eeg source reconstruction, Oct 2016.

[12] Ryan Tibshirani. The lasso problem and uniqueness. Available at url=http://www.stat.cmu.edu/ ryantibs/papers/lassounique.pdf.