ROBUST JOINT SPARSE RECOVERY ON DATA WITH OUTLIERS

Ozgur Balkan[⋆] *Kenneth Kreutz-Delgado*[⋆] *Scott Makeig*[†]

[⋆] Department of Electrical and Computer Engineering
University of California San Diego, La Jolla, CA
[†]Swartz Center for Computational Neuroscience, University of California San Diego, La Jolla, CA

ABSTRACT

We propose a method to solve the multiple measurement vector (MMV) sparse signal recovery problem in a robust manner when data contains outlier points which do not fit the shared sparsity structure otherwise contained in the data. This scenario occurs frequently in the applications of MMV models due to only partially known source dynamics. The algorithm we propose is a modification of MMV-based sparse bayesian learning (M-SBL) by incorporating the idea of least trimmed squares (LTS), which has previously been developed for robust linear regression. Experiments show a significant performance improvement over the conventional M-SBL under different outlier ratios and amplitudes.

Index Terms— Joint Sparse Signal Recovery, Robust Statistics, Sparse Bayesian Learning, Least Trimmed Squares

1. INTRODUCTION

Sparse signal recovery has found a large number of applications in diverse fields of engineering including but not limited to neural networks, telecommunications and biomedical source localization [1]. Typically, the signal model is

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

where $\mathbf{y} \in \mathbb{R}^{M\times 1}$ is the data vector, $\mathbf{A} = [\mathbf{a_1} \dots \mathbf{a_N}] \in \mathbb{R}^{M\times N}$ is the known overcomplete dictionary ($M < N$), $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is the source vector which is assumed to be sparse (number of non zeros values $k < M$), and $\mathbf{e} \in \mathbb{R}^{M\times 1}$ is the noise vector. The goal is to recover the sparse source vector $\mathbf{x}$, given $\mathbf{y}$ and $\mathbf{A}$. It is often sufficient to find the nonzero indices of $\mathbf{x}$, which is called the support set and denoted by S. The nonzero values of $\mathbf{x}$ can be found by solving the undercomplete inverse problem $\mathbf{y} = \mathbf{A_S}\mathbf{x_s}$, where $\mathbf{A_S}$ is the matrix with columns $\mathbf{a_i}$ with $i \in S$. An extension of this problem is provided by the MMV model, which also attracted much attention. In this model, instead of a single data vector $\mathbf{y}$, a data matrix $\mathbf{Y} \in \mathbb{R}^{M\times n}$ is assumed to be generated by the source matrix $\mathbf{X} \in \mathbb{R}^{N\times n}$ as,

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E} \tag{2}$$

where $\mathbf{E} \in \mathbb{R}^{M\times n}$ is the noise vector. The assumption in this model is that the columns of $\mathbf{X}$, denoted by $\mathbf{x_{.i}}$, share a common sparsity structure (joint sparse). That is, it is assumed that nonzero values of $\mathbf{x_{.i}}$ are located at the same rows (indices). Incorporating this assumption results in a significant performance improvement and even more so as the number of measurements n increases [2, 3].

In many applications of MMV algorithms, e.g. source localization for EEG and MEG, the data matrix $\mathbf{Y}$ is obtained by taking out a data window of interest from a larger set of data [4, 5]. If one has a prior knowledge of when the sources turn active and inactive, then the data $\mathbf{Y}$ can be extracted such that the common sparsity assumption holds. However, in most cases this knowledge does not exist and the assumption of common sparsity pattern for 100% of the data becomes far from ideal. In this paper, we refer to data vectors $\mathbf{y_{.i}}$ as outliers if the associated $\mathbf{x_{.i}}$ do not fit the assumed MMV model, which is the shared sparsity assumption.

If the window size n is expanded for performance improvement, the possibility of the data containing outliers increases which can in turn dramatically decrease the algorithm performance. In [6], the MMV algorithms has been shown to be useful for recovering the sparse sources in the case of time-varying sparsity when it is applied on sliding data windows. However, because of the nature of the problem, and unknown locations of sparsity pattern changes, it is rather likely that sliding windows of data contain outliers.

Due to the non-ideal cases described above, we seek an MMV algorithm that is robust to outliers. To do so, we modify the multiple-Sparse Bayesian Learning (M-SBL) algorithm proposed in [3] such that the new algorithm robust-MSBL captures the support set of the majority of data vectors $\mathbf{y_{.i}}$ ignoring the outliers. We adopt the *least trimmed squares* (LTS) method used for robust linear regression in [8, 9] and apply it to the MMV problem. A similar idea is also followed in [10] for robust sparse linear regression and robust PCA [11]. It should be noted that robustness to noise and high sparsity value k is pursued in [12] for the MMV model, however, our approach differs in the sense that we pursue robustness to outliers that are not jointly sparse with the rest of the data.

The outline of the paper is as follows: Section 2 sum-

marizes the M-SBL algorithm and points out the reasons for outlier sensitivity. Section 3 overviews *least trimmed squares* (LTS) and establishes its connection to robust-MSBL. In Section 4, we perform tests. And Section 5 gives some conclusions.

2. M-SBL ALGORITHM

To solve the MMV problem formulated in (2), M-SBL models the rows of the source matrix $\mathbf{X}$ as n-dimensional zero mean gaussian random variables, each having a different covariance matrix controlled by hyperparameters γ_i, $i \in [1, N]$. In other words,

$$p(\mathbf{x_{i.}}; \gamma) \triangleq \mathcal{N}(0, \gamma_i \mathbf{I}), \tag{3}$$

$$p(\mathbf{X}; \gamma) = \prod_{i=1}^{N} p(\mathbf{x_{i.}}; \gamma). \tag{4}$$

$\mathbf{x_{i.}}$ is the i-th row, $\mathbf{x_{.i}}$ is the i-th column of $\mathbf{X}$. Under gaussian noise with a known variance σ^2, we also have $\mathbf{p}(\mathbf{y_{.j}}|\mathbf{x_{.j}}) \triangleq \mathcal{N}(\mathbf{A}\mathbf{x_{.j}}, \sigma^2\mathbf{I})$ with $\mathbf{p}(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{n} \mathbf{p}(\mathbf{y_{.j}}|\mathbf{x_{.j}})$. In [3], M-SBL proceeds with integrating out the unknown sources $\mathbf{X}$ in order to arrive at marginal likelihood of data given the hyper parameters γ, namely $p(\mathbf{Y}; \gamma)$, which is to be maximized. Applying $-2\log()$ transformation we get the M-SBL cost function to be minimized,

$$L(\gamma) \triangleq -2\log(p(\mathbf{Y}; \gamma)) = -2\log \int p(\mathbf{Y}|\mathbf{X}) p(\mathbf{X}; \gamma) d\mathbf{X}$$

$$\equiv \log|\Sigma| + \frac{1}{n} \sum_{t=1}^{n} \mathbf{y_{.t}}^T \Sigma^{-1} \mathbf{y_{.t}} \tag{5}$$

where $\Sigma \triangleq (\mathbf{A}\mathbf{\Gamma}\mathbf{A^T} + \sigma^2\mathbf{I})$, $\Gamma \triangleq \text{diag}(\gamma)$. Although our modifications could be generalized to every method, we will minimize this cost function by using the fixed point update approach, which has significantly faster convergence rate than the EM update [7].

$$\gamma_i^{(k+1)} = \frac{\gamma_i^{(k)}}{\mathbf{a_i^T}(\Sigma^{(k)})^{-1}\mathbf{a_i}} \frac{\|\mathbf{Y^T}(\Sigma^{(k)})^{-1}\mathbf{a_i}\|_2^2}{n} \tag{6}$$

at the $(k+1)$-th iteration and $\gamma_i^{(0)} = 1, \forall i$. The first term $\log|\Sigma|$ in the cost function (5) encourages sparsity of γ whereas the second term tries to fit data as pointed out in [4]. It is also possible to learn the noise parameter σ^2, however it was noted before that the best results are achieved using a fixed value (whether estimated by a different method or using prior knowledge) [3]. After convergence, if the sparsity k is known, one extracts the indices of the largest k values of γ in order to recover the support set.

One can see that all data vectors $\mathbf{y_{.t}}$ and thus $\mathbf{x_{.t}}$ are treated equally in this formulation (has the same weight $\frac{1}{n}$ in (5)), which makes the cost function sensitive to outliers. A large amplitude source outlier $\mathbf{x_{it'}}$ (at time t' in i-th row) is sufficient to boost γ_i since a zero mean gaussian distribution with variance γ_i is fit for i-th row of $\mathbf{X}$. In a scenario where all source vectors $\mathbf{x_{.t}}$ shares the same sparsity pattern this phenomenon does not create a problem in terms of recovering the support set because we would already desire γ_i for $i \in S$ to be large. However, if even one $\mathbf{x_{.t'}}$ that does not share the common sparsity pattern exists (nonzero values at $\mathbf{x_{it'}}$ for $i \notin S$), resulting γ would contain nonzero values at indices $i \notin S$. Moreover, if $\mathbf{x_{it'}}$ for $i \notin S$ is large, the associated γ_i would also be large and thus would be falsely regarded as one of the support set indices.

It can also be observed that in the noiseless limit case, as $\sigma^2 \to 0$, if there exists a data vector $\mathbf{y_{.j}}$ such that $\mathbf{y_{.j}} \notin \mathbf{span}(\mathbf{A_S})$, then any sparse γ satisfying $\gamma_i = 0$ for $i \notin S$, cannot be a local minimum of (5) since $\mathbf{y_{.j}}^T \Sigma^{-1} \mathbf{y_{.j}} \to \infty$. Thus, outliers of the likelihood function (5) are not only the large amplitude data vectors but also the ones that do not share the sparsity pattern of the majority.

3. LTS AND ROBUST-MSBL

3.1. LTS

One of the most common methods for linear regression is *least squares* (LS), where the regression parameter is fit such that the sum of squared residuals are minimized. Equivalently,

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} r_i^2 \tag{7}$$

where θ is the parameter to be optimized and each residual r_i is a function of θ. The weight $\frac{1}{n}$ can be omitted however we keep it to emphasize that what is being minimized is actually the mean of r_i's. Despite its common use, it is known that this method is very sensitive to outliers. Since every point has the same weight $\frac{1}{n}$, a single large outlier can dramatically change the solution. In [8, 9], this problem is analyzed in detail and alternative robust methods are proposed, one of which is *least trimmed squares* (LTS) with the below function to be optimized.

$$\min_{\theta} \sum_{i=1}^{h} (r^2)_{i:n} \tag{8}$$

where $(r^2)_{1:n} \leq (r^2)_{2:n} \leq \ldots \leq (r^2)_{n:n}$ are the ordered residuals, $h \geq \frac{n}{2}$ is the parameter of the LTS estimator determining the number of data points to fit the parameter θ on. LTS therefore allows for some large values of r_i^2 while being able to fit better to the majority of data.

3.2. Robust-MSBL

In order to make M-SBL tolerant to outlier data points, we apply an analogue of the LTS formulation to the conventional

M-SBL. Using the same notation in 3.1, we define the data fit residual for the i-th vector as

$$r_i^2 = \log|\Sigma| + \mathbf{y_{.i}^T}\Sigma^{-1}\mathbf{y_{.i}} \tag{9}$$

which is a function of γ. It can be seen that the conventional M-SBL formulation in (5) is equivalent to the LS estimation when r_i^2 is defined as above. Applying the LTS idea to M-SBL, following cost function is obtained.

$$\begin{aligned} L(\gamma) &= \sum_{t=1}^{h}(r_i^2)_{(t:n)} \\ &= h\log|\Sigma| + \sum_{t=1}^{h}(\mathbf{y_{.i}^T}\Sigma^{-1}\mathbf{y_{.i}})_{(t:n)} \\ &\equiv \log|\Sigma| + \frac{1}{h}\sum_{t=1}^{h}(\mathbf{y_{.i}^T}\Sigma^{-1}\mathbf{y_{.i}})_{(t:n)} \end{aligned} \tag{10}$$

This formulation is equivalent to finding a h-size subset of n columns of $\mathbf{Y}$ that would result in the smallest sum of squared residuals. Given a subset H of size h, we can find γ that minimizes the cost function by conventional M-SBL optimization method given in (6). If $L(\gamma, H)$ denotes the M-SBL objective function restricted to the subset H, we have

$$L(\gamma, H) = \log|\Sigma| + \frac{1}{h}\sum_{t\in H}\mathbf{y_{.t}^T}\Sigma^{-1}\mathbf{y_{.t}} \tag{11}$$

$$\hat{\gamma}_H = \arg\min_{\gamma} L(\gamma, H) \tag{12}$$

and the subset of data which will result in global optimum would be given by

$$H^* = \underset{H\subseteq\{1,2,\ldots,n\},|H|=h}{\arg\min} L(\hat{\gamma}_H, H) \tag{13}$$

This is yet another combinatorial problem where one needs to consider all subsets of size h and perform M-SBL on each of these subsets of data. However, to optimize (10), we follow the iterative method proposed in [9] and also performed in [10]. This method is composed of C-steps which iteratively decrease the objective function value at each step and converge to a local minimum.

3.2.1. C-steps

We start with a random h-size subset of indices $\{1, 2, \ldots, n\}$ and denote this set as H_0. We perform regular M-SBL on this subset of data $\mathbf{Y}$ determined by H_0. With the resulting $\hat{\gamma}_{H_0}$, we compute residuals r_i^2 for all data vectors $\mathbf{y_{.t}}$ as in (9). We find the smallest h of these residuals, assign these indices as the new subset H_1 and keep repeating the same steps until $H_k = H_{k+1}$. As also shown in [10] this method decreases the cost function at each step, as

$$L(\hat{\gamma}_{H_{k+1}}, H_{k+1}) \leq L(\hat{\gamma}_{H_k}, H_{k+1}) \leq L(\hat{\gamma}_{H_k}, H_k) \tag{14}$$

First inequality is valid due to (12) and second inequality is true because of the definition of the next subset H_{k+1}. Of course, it is not guaranteed to reach the global minimum with C-steps due to the random initializations and thus it is best to consider different subset initializations and compare the cost function values obtained at the end. Above verifications are adopted from the sparse LTS regression problem in [10] and applies well to the joint sparse recovery problem. Algorithm 1 presents the pseudocode for robust-MSBL.

Input : $\mathbf{A}, \mathbf{Y}, \sigma^2$, num_initializations, h
Output: γ

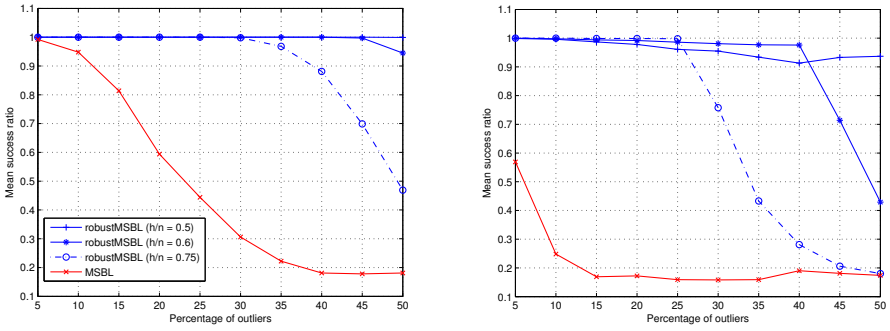
for $trial \leftarrow 1$ **to** *num_initializations* **do**
  $H_0 \leftarrow$ random h-size subset of $\{1, \ldots, n\}$;
  $k \leftarrow 0$;
  repeat
    $\gamma \leftarrow \text{MSBL}(A, Y(:, H_k), \sigma^2)$;
    $\Sigma \leftarrow A\Gamma A^T + \sigma^2 I$;
    for $i \leftarrow 1$ **to** n **do**
      $r_i^2 \leftarrow \log|\Sigma| + \mathbf{y_{.i}^T}\Sigma^{-1}\mathbf{y_{.i}}$;
    end
    $H_{k+1} \leftarrow$ indices of minimum h of r_i^2
    $k \leftarrow k + 1$
  until $H_k = H_{k-1}$;
  costFunc($trial$) $\leftarrow L(\gamma, H_k)$;
  gammas($trial$) $\leftarrow \gamma$;
end
ind $\leftarrow$ index of minimum of costFunc;
$\gamma \leftarrow$ gammas(ind)

Algorithm 1: robust-MSBL

4. EXPERIMENTS

In this section, we perform experiments to compare the performances of MSBL and robust-MSBL under various parameters. We plot the performance with respect to the percentage of the outliers in data, while experimenting with different values of amplitude and sparsity for outliers. At each parameter setting, we perform 100 trials and compute the mean support recovery ratio.

For each trial, we do the following. We create a different random dictionary of size $M = 20$ and $N = 60$ and normalize its columns. We create the source matrix $\mathbf{X}$ of size $N = 60$, $n = 100$ by first separating it into two as $\mathbf{X} = [\mathbf{X_1}; \mathbf{X_2}]$. The first portion of $\mathbf{X}$, i.e. $\mathbf{X_1}$, is the majority of the source vectors which share a common sparsity pattern with sparsity $k_1 = 10$, whereas the second portion of data consists of outliers sharing a different sparsity pattern or not being sparse at all ($k_2 = 10, 60$). Also, note that column permutations of $\mathbf{X}$ or $\mathbf{Y}$ would not affect M-SBL nor robust-MSBL. We randomly select rows with specified parameters k_1 and k_2, and generate source activations from a Gaussian distri-



(a) $k_2 = 10, \frac{\sigma_2}{\sigma_1} = 2$ (b) $k_2 = 10, \frac{\sigma_2}{\sigma_1} = 5$

(c) $k_2 = 60, \frac{\sigma_2}{\sigma_1} = 2$ (d) $k_2 = 60, \frac{\sigma_2}{\sigma_1} = 5$

Fig. 1. $n = 100, k_1 = 10$. More indices (k_2) of γ would be nonzero for (c) and (d) compared to at most $20 = k_1 + k_2$ in (a) and (b), thus lower chance of true support detection for (c) and (d). The higher the outliers' relative amplitude $\frac{\sigma_2}{\sigma_1}$, the higher γ_i for $i \notin S$, hence the performance of correct recovery decreases from (a) to (b) and(c) to (d). As h increases the, performance of robust-MSBL improves since it finds h points to train γ on. However, number of outliers should be lower than $n - h$.

bution $\mathcal{N}(0, \sigma_1)$ for $\mathbf{X_1}$. The outliers $\mathbf{X_2}$ are sampled from $\mathcal{N}(0, \sigma_2)$. We set the parameter $num\_initializations = 1$. Better performance can be achieved using a higher value however a single initialization was sufficient to show the performance improvement over M-SBL. We add noise to the simulated data such that SNR = 10dB. We recover the support set $\hat{S}$ by extracting the indices of the largest k_1 values of γ's returned by the algorithms and compute the mean success ratio as $\frac{1}{100}\sum_{\text{trial}=1}^{100} |\hat{S} \cap S|/k_1$. Figure 1 shows the results when data contains outliers. Figure 2 shows the results when there are no outliers.

5. CONCLUSION

In this work, we modified M-SBL [3] by exploiting the idea of least trimmed squares. This modification significantly enhances MSBL's robustness to data which contain outliers not sharing the common data sparsity structure. Experiments demonstrate that this approach outperforms M-SBL in recovering the correct support set.

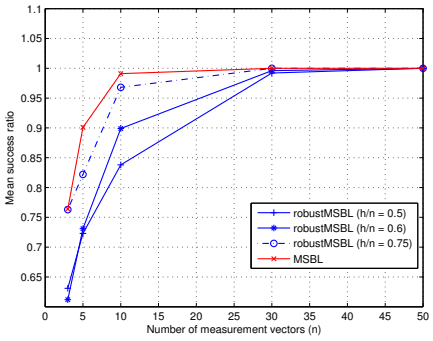


Fig. 2. Data with no outliers (ideal case). As expected, if n is small, M-SBL performs better than robust-MSBL when there are no outliers because robust-MSBL finds the best $h < n$ data vectors to optimize γ on, whereas M-SBL uses all n data vectors. However, if the data window n is large enough ($n > 30$ for this experiment), h points become sufficient for robust-MSBL to be as successful as M-SBL.

6. REFERENCES

[1] R.G. Baraniuk, E. Candes, M. Elad, and Y. Ma, "Applications of sparse representation and compressive sensing [scanning the issue]," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 906–909, 2010.

[2] Shane F. Cotter, Bhaskar D. Rao, Kjersti Engan, Kenneth Kreutz-delgado, and Senior Member, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Processing*, pp. 2477–2488, 2005.

[3] D.P. Wipf and B.D. Rao, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3704 –3716, july 2007.

[4] David P. Wipf, Julia P. Owen, Hagai Attias, Kensuke Sekihara, and Srikantan S. Nagarajan, "Estimating the location and orientation of complex, correlated neural activity using meg," in *NIPS*, 2008, pp. 1777–1784.

[5] Karl Friston, Lee Harrison, Jean Daunizeau, Stefan Kiebel, Christophe Phillips, Nelson Trujillo-Barreto, Richard Henson, Guillaume Flandin, and Jrmie Mattout, "Multiple sparse priors for the m/eeg inverse problem," *NeuroImage*, vol. 39, no. 3, pp. 1104 – 1120, 2008.

[6] Z. Zhang and B. D. Rao, "Exploiting Correlation in Sparse Signal Recovery Problems: Multiple Measurement Vectors, Block Sparsity, and Time-Varying Sparsity," *ArXiv e-prints*, May 2011.

[7] D.P. Wipf, *Bayesian Methods For Finding Sparse Representations*, Ph.D. thesis, UC San Diego, 2006.

[8] P.J. Rousseeuw, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[9] Peter J. Rousseeuw and Katrien Driessen, "Computing LTS regression for large data sets," *Data Min. Knowl. Discov.*, vol. 12, no. 1, pp. 29–45, Jan. 2006.

[10] A. Alfons, C. Croux, and S. Gelper, "Sparse least trimmed squares regression," *Available at SSRN 1967418*, 2011.

[11] D.A. Jackson and Y. Chen, "Robust principal component analysis and outlier detection with ecological data," *Environmetrics*, vol. 15, no. 2, pp. 129–139, 2004.

[12] M.M. Hyder and K. Mahata, "A robust algorithm for joint-sparse recovery," *Signal Processing Letters, IEEE*, vol. 16, no. 12, pp. 1091 –1094, dec. 2009.