Fast Estimation of Correlated Sparse Vectors Via Sparse Bayesian Learning

Nitin Gupta (190567)

***Abstract*—Correlated Sparse Bayesian learning (SBL-CORR) is a powerful framework for tackling the sparse coding problem which exploits both the underlying spatial sparsity as well as the spatial correlation in the channel. As in practice, this algorithm becomes too expensive for high dimensional problems. This is due to the inverse computation a large covariance matrix. To address this problem, we exploit the properties of the covariance matrix and thus propose a fast novel SBL-CORR-COFEM algorithm which locates the non zero entries in the inverse of covariance matrix, and then estimates the inverse in much lesser time complexity by solving multiple linear systems. Our results show that, when compared to a genie-aided estimator LMMSE and SBL-CORR, our algorithm gives the result in much lesser time complexity with negligible difference in performance.**

***Index Terms*—Correlated Sparse Bayesian learning (SBL-CORR), Covariance Free Correlated Sparse Bayesian learning (SBL-CORR-COFEM), Covariance Free Expectation Maximisation (CoFEM)**

I. Introduction

SPARSE signal recovery has received considerable attention due to their desirable capacity of improved performance in signal detection. The sparse signal recovery problem can be modelled as:

$$\mathbf{y}_r = \mathbf{\Phi}_r \mathbf{g}_r + \mathbf{n}_r, \tag{1}$$

where, $\mathbf{y}_r \in \mathbb{R}^{M\times 1}$ is the observed signal, $\mathbf{\Phi}_r \in \mathbb{R}^{M\times N}$ is the known dictionary matrix, $\mathbf{g}_r \in \mathbb{R}^{N\times 1}$ is the sparse signal which needs to be estimated. It has a zero mean and covariance matrix $\mathbf{R_g}$. The term $\mathbf{n}_r \in \mathbb{R}^{M\times 1}$ is the noise vector with a known noise variance. We assume that the sparse signal $\mathbf{g}_r$ is correlated and model it using the uniform correlation model, with ρ_{ij} denoting the correlation coefficient between the ith and jth entries. The variance of i^{th} entry of $\mathbf{g}_r$ is denoted as γ_i^*. If the ith entry o f$\mathbf{g}_r = 0$, the variance $\gamma_i^* = 0$. The covariance between the (i, j) th entries of the channel $\mathbf{g}_r$ is given as $[\mathbf{R_g}]_{ij}$ = $\rho_{ij}\sqrt{\gamma_i^*}\sqrt{\gamma_j^*}$. Let the matrix $\Gamma^* = \text{diag}(\gamma_1^*, \gamma_2^*, ..., \gamma_G^*)$. The covariance matrix is given as $\mathbf{R_g} = (\Gamma^*)^{1/2}\mathbf{U}(\Gamma^*)^{1/2}$, where the correlation matrix is given by $\mathbf{U} \in \mathbb{C}^{D\times D}$ with ρ_{ij} as its (i, j)th entry. Our goal is to exploit sparsity and correlation in the channel and thus estimate it using Sparse Bayesian Learning.

Tipping in [1] developed the Bayesian learning framework for calculating sparse solutions to regression and classification tasks. The framework uses an expectation maximization procedure to learn the parameters of the prior, and update the posterior estimates. Lee *et al.* in [2] discussed different sparse representations of the mmWave channel. Their design applied orthogonal matching pursuit (OMP) to estimate the channel using a reduced number of pilots. Prasanna and Murthy in [3] exploited both sparsity and correlation of the sparse vector, while estimating it. They proposed a Gaussian prior model which captures the intra-vector correlation. Exploiting both sparsity and correlation improves their estimation performance. The algorithm in [3], however, inverts a high-dimensional covariance matrix to update the posteriors. This radically increases their algorithm complexity. Zhilin Zhang and Bhaskar D. Rao in [4] discussed Block Sparse Bayesian learning (BSBL) framework for estimating the block-structured sparse signals. Most popular BSBL algorithms become too expensive for high dimensional problems. This is due to the inverse computation of a large covariance matrix. Our algorithm addresses the problem of all such algorithms.

In this work, we address this complexity problem by bypassing the matrix inversion. The proposed approach provide a low-complexity solution for estimating correlated sparse vectors. The main **contributions** of this work which help in achieving this objective can be summarized as follows.

1) We propose a covariance-free expectation maximization algorithm for correlated SBL (SBL-CORR) algorithm proposed in [3]. The proposed algorithm provides a low time complexity solution with a marginal degradation in the normalized mean square error. We accomplish this by reducing the time taken to calculate inverse of a high-dimensional matrix by exploiting the sparsity in the inverse of matrix, and by using ideas from linear algebra.

2) We analytically show that the inverse of covariance matrix is sparse, and its structure is indeed same as the correlation matrix. We then identify locations of non zero entries in the covariance matrix and thus estimate the vector sparsity, which are exploited to reduce the dimension of linear system of equations used to obtain the inverse of covariance matrix.

II. Existing Inference Schemes For SBL-CORR

In this section, we review the main ideas behind the design in [3], and comment on some of the shortcomings in recovering high-dimensional vectors $\mathbf{z}$ for large D.

The sparse bayesian algorithm in [3] involves two steps: obtaining the optimal value for the hyper parameters $\mathbf{c}$, and computing the posterior distribution. We consider a parameterized Gaussian prior for $\mathbf{g}_r$ as:

$$p(\mathbf{g}_r; \mathbf{c}) = \mathcal{CN}(\mathbf{g}_r; \mathbf{0}, \mathbf{\Omega_c}^{-1}), \tag{2}$$

where, vector $\mathbf{c} \in \mathbb{R}_+^D$ has $1/\sqrt{\gamma_i}$ as ith entry. The precision matrix $\mathbf{\Omega_c} = \mathbf{C}\mathbf{U}^{-1}\mathbf{C}$ where $\mathbf{C} = diag(\mathbf{c})$. This choice of

The authors are with the Department of Electrical Engineering, IIT Kanpur, Kanpur 208016, India. email: {niting, mailidsecondauthor, mailidthirdauthor}@iitk.ac.in.

prior model is known to induce sparsity in the final channel estimate. Then, the posterior distribution of $\mathbf{g}_r$ given the observation $\mathbf{y}_r$ and hyper parameter $\mathbf{c}$, is

$$p(\mathbf{g}_r|\mathbf{y}_r;\mathbf{c}) = \mathcal{CN}(\mathbf{g}_r;\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}},\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}), \tag{3}$$

where,

$$\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}} = \frac{1}{\sigma_n^2}\boldsymbol{\Phi}^H\boldsymbol{\Phi} + \boldsymbol{\Omega}_{\mathbf{c}}, \tag{4}$$

$$\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}} = \frac{1}{\sigma_n^2}\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}\boldsymbol{\Phi}^H\mathbf{y}, \tag{5}$$

Since the posterior distribution of $\mathbf{g}_r$ is Gaussian, its mode is the same as its mean $\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}} \in \mathbb{C}^{D\times 1}$ with covariance matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}} \in \mathbb{C}^{D\times D}$. The cost function that needs to be maximized for finding $\mathbf{c}$ is obtained from the log likelihood $log(p(\mathbf{y}_r;\mathbf{c},\sigma_n^2))$ as

$$L(\mathbf{c}) = logdet(\boldsymbol{\Omega}_{\mathbf{y}}) - \mathbf{y}_r^H\boldsymbol{\Omega}_{\mathbf{y}}\mathbf{y}_r, \tag{6}$$

The problem of maximizing the cost function (6)is nonconvex and does not admit a closed form solution. We use EM approach which gives the following updates for $\mathbf{c}$ and $\hat{\mathbf{R}}_{\mathbf{g}}$

$$\hat{\mathbf{R}}_{\mathbf{g}} = \boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1} + \boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}^H, \tag{7}$$

$$\mathbf{c}_{new} = (Re\{\mathbf{U}^{-1}\odot\hat{\mathbf{R}}_{\mathbf{g}}^T\})^{-1}\frac{1}{\mathbf{c}_{old}}, \tag{8}$$

The E-step of algorithm updates $\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}$ and $\hat{\boldsymbol{\Omega}}_{\mathbf{g}|\mathbf{y}}$ via equations (5) and (4) respectively and M-step of algorithm updates $\hat{\mathbf{R}}_{\mathbf{g}}$ and $\mathbf{c}$ via equations (7) and (8) respectively. Thus the iterative algorithm to estimate the sparse vector $\mathbf{g}_r$ from [3] can be articulated as follows:

Algorithm 1: Corr-SBL algorithm.

Input: $\boldsymbol{\Phi}, \{\mathbf{y}\}_{r=1}^T, \mathbf{U}, \sigma_n^2$
Output: $\{\hat{\mathbf{g}}_r\}_{r=1}^T$
1 **while** $k \le k_{max}$ and $|||[\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}]_k - [\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}]_{k-1}||_F < \epsilon$ **do**
2 $\quad\hat{\mathbf{R}}_{\mathbf{g}} = \mathbf{0}_{D\times D}$
3 $\quad\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}} = \frac{1}{\sigma_n^2}\boldsymbol{\Phi}^H\boldsymbol{\Phi} + \boldsymbol{\Omega}_{\mathbf{c}}$
4 $\quad\boldsymbol{\Omega}\_\mathbf{inv} = \boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$
5 $\quad\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}} = \frac{1}{\sigma_n^2}\ \boldsymbol{\Omega}\_\mathbf{inv}\ \boldsymbol{\Phi}^H\mathbf{y}$
6 $\quad$**for** r ← 1 to T **do**
7 $\quad\quad\hat{\mathbf{R}}_{\mathbf{g}} \leftarrow \hat{\mathbf{R}}_{\mathbf{g}} + \frac{[\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}]_{([D],r)}[\boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}]_{([D],r)}^H}{T}$
8 $\quad\hat{\mathbf{R}}_{\mathbf{g}} \leftarrow \hat{\mathbf{R}}_{\mathbf{g}} + \boldsymbol{\Omega}\_\mathbf{inv}$
9 $\quad\mathbf{c} \leftarrow (Re\{\mathbf{U}^{-1}\odot\hat{\mathbf{R}}_{\mathbf{g}}^T\})^{-1}\frac{1}{\mathbf{c}}$
10 $\quad\boldsymbol{\Omega}_{\mathbf{c}} = diag(\mathbf{c})\mathbf{U}^{-1}diag(\mathbf{c})$
11 $\quad k \leftarrow k+1$
12 **return** $\{\hat{\mathbf{g}}_r\}_{r=1}^T = \mu_{\mathbf{g}|\mathbf{y}}, \mathbf{c}.$

The algorithm takes the highest time in Step 4 where it calculates the inverse of $D \times D$ matrix. It takes $O(D^3)$ time in every iteration. We also note that this inverse is required in Steps 5 and 8. We can avoid the use of $\boldsymbol{\Omega}\_\mathbf{inv}$ by solving the linear system in Step 5 in lesser time complexity. Assuming $\mathbf{A} = \boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}$, $\mathbf{X} = \boldsymbol{\mu}_{\mathbf{g}|\mathbf{y}}$ and $\mathbf{B} = \frac{1}{\sigma_n^2}\boldsymbol{\Phi}^H\mathbf{y}$, the system $\mathbf{AX} = \mathbf{B}$ can be solved in $O(T \times D^2)$ time which is much better than $O(D^3)$ time taken earlier. One option to solve is this system is by using the Conjugate Gradient method [5]. The authors of [5] proposed new inference algorithm for simple SBL called covariance free expectation-maximization (CoFEM). CoFEM accelerates the EM algorithm by eliminating the covariance matrix inversion step. The simple EM algorithm has the update $\hat{\boldsymbol{\alpha}}^{new} = 1 \oslash (\boldsymbol{\mu}\odot\boldsymbol{\mu} + trace(\boldsymbol{\Sigma}))$, in the M-step, where $\boldsymbol{\mu} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{y}$ and $\boldsymbol{\Sigma} = (\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} + diag\{\hat{\boldsymbol{\alpha}}\})^{-1}$. The authors of [5] observed that not all elements of $\boldsymbol{\Sigma}$ are required for the M-step. We only need the diagonal elements of $\boldsymbol{\Sigma}$. To estimate the diagonal entries they used a result from [6]. But the Corr-SBL algorithm in Step 8, unlike COFEM, also requires calculation of non-diagonal entries of $\boldsymbol{\Omega}\_\mathbf{inv}$ to calculate $\hat{\mathbf{R}}_{\mathbf{g}}$. The COFEM algorithm is thus not directly applicable here. We now develop a novel technique to invert the matrix in a lesser time complexity.

III. Proposed design

The proposed algorithm aims to speed up the Corr-SBL algorithm by exploiting the properties of covariance matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$. We leverage tools from the numerical linear algebra literature to accomplish this goal. We first state the following lemma.

Lemma 1. The matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ is sparse .

Proof. See Appendix A. □

Our main challenge is to estimate the $D \times D$ inverse in lesser than $O(D^3)$ time. To achieve this objective, we state the following lemma:

Lemma 2. Assume a $D \times D$ matrix $\mathbf{M}$, which has a sparse inverse, and known locations of non zero entries. We can obtain the inverse of matrix M by solving the following linear system of equations.

$$\mathbf{M} \times \mathbf{X} = \mathbf{B}$$

The non sparse columns of inverse of $\mathbf{M}$ can be obtained by solving the above linear system for $\mathbf{X} \in \mathbb{C}^{D\times L}$, where L is the number of non sparse columns of $\mathbf{M}^{-1}$. The matrix $\mathbf{B} \in \{0,1\}^{D\times L}$ contains L columns of identity matrix corresponding to the non sparse columns of $\mathbf{M}^{-1}$. As we know the location of non sparse columns in matrix $\mathbf{M}^{-1}$, using the solution $\mathbf{X}$, we can obtain the entire matrix $\mathbf{M}^{-1}$ by simply substituting the columns of $\mathbf{X}$ to their respective places in $\mathbf{M}^{-1}$ and keeping other columns all zeros.

Proof. See Appendix B. □

To reduce the time complexity of SBL-CORR, we need to reduce the time taken by step 4 in the algorithm. According to Lemma 1, the matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ is sparse. To estimate this inverse using Lemma 2, we need to also obtain the location of non zero entries in $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$. Calculating the inverse of whole matrix takes $O(D^3)$ time. But using Lemma 2, we can reduce the dimension of linear system and solve it in $O(LD^2)$ time which is much better than the conventional way.

For obtaining the location of non zero columns in matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$, we perform 10 iterations of SBL-CORR to obtain a crude estimate of the inverse $\boldsymbol{\Omega}_{\mathbf{inv}}$ which involves inversion of matrix. In simulations we show that even after performing 10 iterations of complex computations, we can obtain a significant improvement in speed in our algorithm. From this crude estimate of $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ we can locate the non zero columns in $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ using **cal_L** algorithm described below.

As the matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ is sparse, some columns will contain non-zero entries and others will be all zeros. We need to locate the columns which contain non-zero entries. To reduce this 2-D problem to 1-D, we sum up the rows of matrix $\boldsymbol{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$. This will give us a vector $\boldsymbol{\Omega}\_\mathbf{inv}\_\mathbf{sum}$ with zero entries

corresponding to the columns which were all zeros and non-zeros entries for others.

$$\mathbf{\Omega\_inv\_sum} = \mathbf{\Sigma}_{i=1}^{D} |\mathbf{\Omega_{inv}}(i,:)|$$

Now, we can easily select the indices of entries in $\mathbf{\Omega\_inv\_sum}$ which are atleast 15% of highest entry in $\mathbf{\Omega\_inv\_sum}$. This gives us the location of non-zero columns in matrix $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$. And the number of such columns gives us the sparsity L. This algorithm can be summarised as follows:

Algorithm 2: *cal_L*

Input: $\mathbf{\Omega\_inv}$
Output: L, ind
$\mathbf{\Omega\_inv\_sum} = \mathbf{\Sigma}_{i=1}^{D} abs(\mathbf{\Omega\_inv}(i,:))$
$\alpha \leftarrow$ max entry in $\mathbf{\Omega\_inv\_sum}$
$L \leftarrow 0$
$k \leftarrow 0$
for $i \leftarrow 1$ to D **do**
  if $\mathbf{\Omega\_inv\_sum}(i) > 0.15 * \alpha$ **then**
    $L \leftarrow L + 1$
    $ind[k] \leftarrow i$
    $k \leftarrow k + 1$
return L, ind

Calculating inverse of matrix:
As we have located the non zero columns of $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$, we can use Lemma 2 to calculate this inverse in $O(LD^2)$ time. We build the matrix $\mathbf{B}$ as described in the lemma using the vector **ind** which stores the location of non zero columns in $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$. Then we solve for $\mathbf{X}$ and obtain the solution for covariance matrix $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$.

Algorithm 3: getInv

Input: $\mathbf{\Omega}_{\mathbf{g|y}}, D, L, ind$
Output: $\mathbf{\Omega\_inv}$
$\mathbf{B} = \mathbf{0}_{D \times L}$
for i = 1 to L **do**
  $\mathbf{B}[ind(i), i] = 1$
$\mathbf{X} = linearsolver(\mathbf{\Omega}_{\mathbf{g|y}}, \mathbf{B})$
$\mathbf{\Omega\_inv} = \mathbf{0}_{D \times D}$
for i = 1 to L **do**
  $\mathbf{\Omega\_inv}[:, ind(i)] = \mathbf{X}(:, i)$
return $\mathbf{\Omega\_inv}$

Hence SBL-CORR-COFEM algorithm can be given as follows:

Algorithm 4: Corr-SBL-2

Input: $\mathbf{\Phi}, \{\mathbf{y}\}_{r=1}^{T}, \mathbf{U}, \sigma_n^2$
Output: $\hat{\mathbf{g}}_r$
while $k \leq k_{max}$ and $|||[\boldsymbol{\mu}_{\mathbf{g|y}}]_k - [\boldsymbol{\mu}_{\mathbf{g|y}}]_{k-1}||_F < \epsilon$ **do**
  $\hat{\mathbf{R}}_{\mathbf{g}} = \mathbf{0}_{D \times D}$
  if $k < 10$ **then**
    $\mathbf{\Omega}_{\mathbf{g|y}} = \frac{1}{\sigma_n^2} \mathbf{\Phi}^H \mathbf{\Phi} + \mathbf{\Omega}_{\mathbf{c}}$
    $\mathbf{\Omega\_inv} = \mathbf{\Omega}_{\mathbf{g|y}}^{-1}$
    $\boldsymbol{\mu}_{\mathbf{g|y}} = \frac{1}{\sigma_n^2} \mathbf{\Omega\_inv}\, \mathbf{\Phi}^H \mathbf{y}$
    $L \leftarrow cal\_L(\mathbf{\Omega\_inv})$ //This function counts the number of non sparse columns in $\mathbf{\Omega\_inv}$
  else
    $\mathbf{\Omega}_{\mathbf{g|y}} = \frac{1}{\sigma_n^2} \mathbf{\Phi}^H \mathbf{\Phi} + \mathbf{\Omega}_{\mathbf{c}}$
    $\mathbf{\Omega\_inv} \leftarrow getInv(\mathbf{\Omega}_{\mathbf{g|y}}, \mathbf{\Omega\_inv}, D, L)$ //getInv gives estimate of inverse in $O(LD^2)$ time
    $\boldsymbol{\mu}_{\mathbf{g|y}} = \frac{1}{\sigma_n^2} \mathbf{\Omega\_inv}\, \mathbf{\Phi}^H \mathbf{y}$
  for r $\leftarrow$ 1 to T **do**
    $\hat{\mathbf{R}}_{\mathbf{g}} \leftarrow \hat{\mathbf{R}}_{\mathbf{g}} + \frac{[\boldsymbol{\mu}_{\mathbf{g|y}}]_{([D],r)} [\boldsymbol{\mu}_{\mathbf{g|y}}]_{([D],r)}^H}{T}$
  $\hat{\mathbf{R}}_{\mathbf{g}} \leftarrow \hat{\mathbf{R}}_{\mathbf{g}} + \mathbf{\Omega\_inv}$
  $\mathbf{c} \leftarrow (Re\{\mathbf{U}^{-1} \odot \hat{\mathbf{R}}_{\mathbf{g}}^T\})^{-1} \frac{1}{\mathbf{c}}$
  $\mathbf{\Omega}_{\mathbf{c}} = diag(\mathbf{c}) \mathbf{U}^{-1} diag(\mathbf{c})$
  $k \leftarrow k + 1$
return $\{\hat{\mathbf{g}}_r\}_{r=1}^{T} = \boldsymbol{\mu}_{\mathbf{g|y}}, \mathbf{c}$.

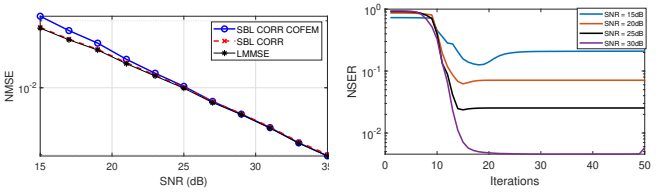
In this algorithm all updates are same as SBL_CORR except the calculation of $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$ after 10 iterations. First of all we do 10 iterations of $CORR\_SBL$ algorithm to get an estimate of $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$. After these iterations we can use $getInv$ algorithm to obtain the inverse in $O(LD^2)$ time.

IV. Time Complexity Analysis

The time complexity of $getInv$ Algorithm 3 is $O(LD^2)$. This is mainly due to the 6^{th} step which solves a linear system of equation. The time complexity of cal_L Algorithm 2 is $O(LD^2)$. This is mainly due to the 1st step, which sums up the rows of $\mathbf{\Omega\_inv}^{old}$. For $SBL - CORR - 2$ we are doing initial 10 iterations of SBL_CORR which involves inversion of a $D \times D$ matrix taking $O(D^3)$ time. And for the subsequent steps $(K_{max} - 10)$ we calculate the inverse in $O(LD^2)$ time using $getInv$ algorithm. The time complexity of original SBL_CORR algorithm was $O(K_{max} D^3)$. Thus the total time complexity of $SBL - CORR - 2$ algorithm is $O(10D^3 + (K_{max} - 10)LD^2)$. The proposed algorithm has one less complexity order than conventional $SBL - CORR$ algorithm.

V. Simulation Results

We now numerically investigate the performance of the proposed SBL-CORR-COFEM algorithm. We compare its performance with the SBL-CORR algorithm in [3], which is designed for estimating the mmWave channel and prove that the time complexity of our algorithm SBL-CORR-COFEM is much lesser than SBL-CORR with a negligible difference in performance. We also plot the oracle LMMSE, which knows the true user activity, and lower bound the performance of two algorithms. We first compare the normalized mean square error (NMSE) of these algorithms which is defined as $\mathbb{E}[||\hat{\mathbf{w}} - \mathbf{w}||^2 / ||\mathbf{w}||^2]$, where $\hat{\mathbf{w}}$ is the estimate of $\mathbf{w}$. For our experiments, we consider a BS with $N = 128$ antennas which serves $K = 10$ single-antenna users. The BS employs a hybrid MIMO architecture with $M = 60$ RF chains. The channels have uniform correlation with ρ_{ij} = 0.5. The entries of the design matrix $\mathbf{\Phi}$ are drawn from independent complex normal $\mathcal{CN}(0,1)$ distribution. The maximum number of iterations $k_{max} = 400$. The threshold value ϵ is set as 10^{-3}. This value is kept the same across all algorithms.



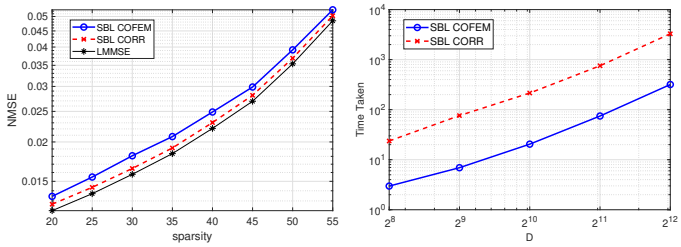
(a) (b)

Fig. 1: (a) Plot comparing SBL_CORR and SBL_CORR_2 vs noise variance; (b) Plot comparing NSER vs iterations for various noise variances for matrix $\mathbf{\Omega}_{\mathbf{g|y}}^{-1}$ calculated by SBL-CORR-2

We first compare in Fig.1a, the NMSE of three algorithms by varying the SNR. We see that for $SNR \leq 20$ dB, the NMSE of the proposed SBL-CORR-COFEM algorithm is marginally higher than the SBL-CORR algorithm. But for

$SNR > 20$ dB, both of them have a similar NMSE. This is because at low SNR values, the high noise makes the inverse matrix erroneous which makes it non-sparse. From Fig.1b, we infer that at high SNR values, our assumption regarding the sparsity of inverse of $\mathbf{\Omega_{g|y}}$ is valid as NSER is very low for SNR = 30dB.

We now fix SNR $=$ 25 dB and investigate the NMSE by varying other parameters. We plot the performance by averaging it 100 channel realizations and T = 50 coherent blocks.

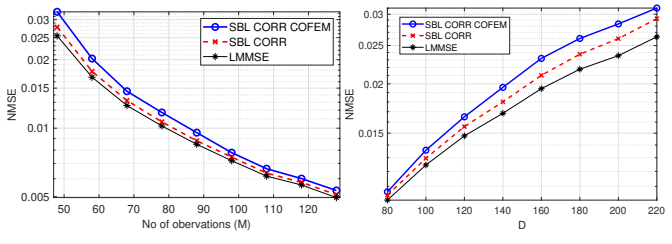


(a) (b)

Fig. 2: (a) Plot comparing NMSE performance of SBL_CORR and SBL_CORR_2 against Sparsity level; (b) Plot comparing time taken by SBL_CORR and SBL_CORR_2 to converge within 0.001. All times are recorded on CPUs.

Figure 2a compares the NMSE by varying the sparsity. We see that the NMSE of all the three algorithms increase with sparsity level. This is because with increase in sparsity, the number of non-zero entries which needs to be estimated increase, but the number of observations remain same. This degrades their performance. We again see that the NMSE of the proposed SBL-CORR-COFEM algorithm is only marginally higher than the SBL-CORR algorithm.

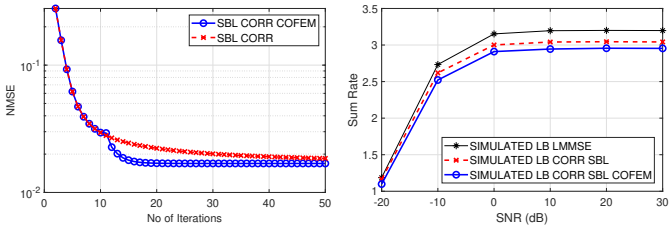
Figure 2b compares the time complexity the proposed algorithm with SBL-CORR. We see that the proposed algorithm is $10-12$ times faster than SBL-CORR for all values of grid size D. This is because SBL-CORR involves taking inverse of covariance matrix $\mathbf{\Omega}_{g|y}$ in step 4, which is a $D \times D$ matrix. As D is very large, this step takes $O(D^3)$ time. On the other hand, our algorithm bypasses this step by estimating the inverse of this matrix in lesser time complexity. Thus, the proposed algorithm is $10-12$ times faster with a similar NMSE performance.



(a) (b)

Fig. 3: (a) Plot comparing SBL_CORR and SBL_CORR_2 vs number of observations; (b) Plot comparing SBL_CORR and SBL_CORR_2 vs dimension of matrix.

Figure 3a compares the NMSE of algorithms by varying the number of observations M for a fixed sparsity level. We see that with increase in M, the NMSE of all the three algorithms improve. This is because with increase in M the algorithm gets extra information while the unknown sparsity level remains fixed. This increases the algorithm accuracy. We again note that the NMSE of three algorithms is almost similar. Figure 3b compares the NMSE by varying D and by keeping other variables constant. We see that with increase in grid size D, the NMSE of all the three algorithms degrades. This is because the number of unknowns increases but the number of observations is constant. Hence we see an increase in NMSE for all the algorithm. We again note that the NMSE of three algorithms is almost similar.



(a) (b)

Fig. 4: (a) Plot comparing the convergence of SBL_CORR and SBL_CORR_2 vs number of iterations; (b) Plot comparing the spectral efficiency of SBL_CORR and SBL_CORR_2 vs the noise variance.

We show in Fig. 4a the convergence of both the algorithms. We see that that the proposed SBL-CORR-COFEM algorithm converges faster. In SBL-CORR-COFEM, we are estimating only the non-zero entries in the inverse of covariance matrix $\mathbf{\Omega}^{-1}_{\mathbf{g|y}}$. We assume other entries of this matrix to be exactly equal to zero. We are thus reducing the noise in this matrix. Whereas in SBL-CORR, it takes many iterations for these entries to converge to zero recursively. The SBL-CORR-COFEM algorithm converges within $25-30$ iterations, whereas SBL-CORR requires more than 200 iterations.

We compare in Fig. 4b, the spectral efficiency (SE) of three algorithms by averaging for 100 channel realizations, T = 50 coherent blocks and $K = 10$ single-antenna users. The SE metric characterizes the tangible performance of a wireless system. Let $\hat{\mathbf{h}}_i$ be the channel estimate for ith user and $\hat{\mathbf{H}}$ = $[\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2...\hat{\mathbf{h}}_K]$. Suppose kth user transmits a symbol x_k with zero mean and power P. Then the received combined vector at the base station is given by:

$$\mathbf{y} = \mathbf{F}\mathbf{W}_{RF}\left(\sum_{k=1}^{K}\mathbf{h}_k x_k + \mathbf{n}\right), \tag{9}$$

where $\mathbf{W}_{RF} \in \mathbb{C}^{M\times N}$ is the analog combiner matrix. The digital combiner is given by:

$$\mathbf{F} = (\hat{\mathbf{H}}_B^H\hat{\mathbf{H}}_B + \eta\mathbf{I}_M)^{-1}\hat{\mathbf{H}}_B^H \in \mathbb{C}^{M\times M}, \tag{10}$$

where effective baseband channel is denoted by $\hat{\mathbf{H}}_B = \mathbf{W}_{RF}\hat{\mathbf{H}}$ and $\eta = \frac{M\sigma_n^2}{P}$ is the regularisation parameter. For computing the spectral efficiency, we write the kth row of (9) as

$$y_k = \mathbb{E}[\mathbf{f}_k\mathbf{W}_{RF}\mathbf{h}_k]x_k + (\mathbf{f}_k\mathbf{W}_{RF}\mathbf{h}_k x_k - \mathbb{E}[\mathbf{f}_k\mathbf{W}_{RF}\mathbf{h}_k]x_k) + \sum_{q=1,q\neq k}^{K}\mathbf{f}_k\mathbf{W}_{RF}\mathbf{h}_q x_q + \mathbf{f}_k\mathbf{W}_{RF}\mathbf{n} \tag{11}$$

The term $\mathbb{E}[\mathbf{f}_k\mathbf{W}_{RF}\mathbf{h}_k]$ is treated as a known deterministic

channel. Then using a result from [3], we get

$$SE_k \geq \left(1 - \frac{K}{\tau_c}\right) \log_2(1 + \widetilde{\gamma}_k), \tag{12}$$

where the pre-log factor accounts for pilot overhead, and $\widetilde{\gamma}_k$ is an effective SINR term given by

$$\widetilde{\gamma}_k = \frac{P|\mathbb{E}[\mathbf{f}_k \mathbf{W}_{RF} \mathbf{h}_k]|^2}{P\sum_{q=1}^{K} \mathbb{E}[|\mathbf{f}_k \mathbf{W}_{RF} \mathbf{h}_k|^2] - P|\mathbb{E}[\mathbf{f}_k \mathbf{W}_{RF} \mathbf{h}_k]|^2 + \alpha_k}, \tag{13}$$

where $\alpha_k = \sigma_n^2 \mathbb{E}[\mathbf{f}_k \mathbf{W}_{RF} \mathbf{W}_{RF}^H \mathbf{f}_k^H]$ is the effective noise power in the combined signal. We see that all the three algorithms have a similar SE. This plots shows that the proposed design makes the algorithm faster, but without degrading the practical performance.

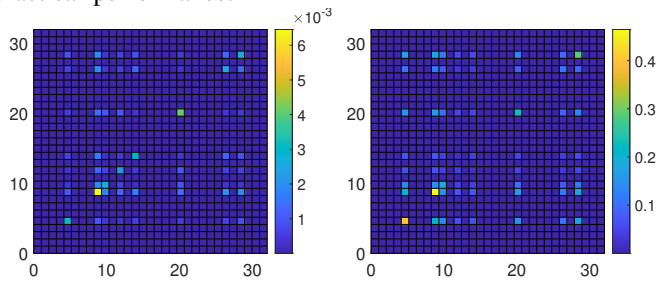


Fig. 5: (a) Matrix $\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ obtained from SBL-CORR-2; (b) Matrix $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ obtained from SBL-CORR-2

Figure 5a and 5b compares the structure of matrices $\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ and $\mathbf{\Omega}_{\mathbf{c}}^{-1}$. We infer that the non zero locations of both the matrices is same and they are indeed sparse.

VI. Conclusion

In this paper, we developed a faster algorithm to accelerate SBL-CORR algorithm. By leveraging tools from linear algebra and exploiting sparsity, we solved the linear system of equations in lesser time complexity and by-passed taking inverse of high dimensional matrix. By measuring the channel estimation performance, we prove that without sacrificing much of reconstruction performance, we can estimate the sparse channel using the proposed algorithm in $O(LD^2)$ time. Whereas, SBL-CORR algorithm takes $O(D^3)$ time for every iteration.

Appendix

A. Proof of Lemma 1

We have from **(??)**

$$\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}} = \frac{1}{\sigma_n^2} \mathbf{\Phi}^H \mathbf{\Phi} + \mathbf{\Omega}_{\mathbf{c}}$$

Using Woodbury Identity **[?]**, we have

$$\begin{aligned} \mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1} &= \mathbf{\Omega}_{\mathbf{c}}^{-1} - \mathbf{\Omega}_{\mathbf{c}}^{-1} \mathbf{\Phi}^H (\sigma^2 \mathbf{I} + \mathbf{\Phi} \mathbf{\Omega}_{\mathbf{c}}^{-1} \mathbf{\Phi}^H)^{-1} \mathbf{\Phi} \mathbf{\Omega}_{\mathbf{c}}^{-1} \\ &= \mathbf{\Omega}_{\mathbf{c}}^{-1} - \mathbf{\Omega}_{\mathbf{c}}^{-1} \mathbf{A} \mathbf{\Omega}_{\mathbf{c}}^{-1}. \end{aligned}$$

Here $\mathbf{A} = \mathbf{\Phi}^H (\sigma^2 \mathbf{I} + \mathbf{\Phi} \mathbf{\Omega}_{\mathbf{c}}^{-1} \mathbf{\Phi}^H)^{-1} \mathbf{\Phi}$. We also have

$$\begin{aligned} \mathbf{\Omega}_{\mathbf{c}} &= \text{diag}(\mathbf{c}) \mathbf{U}^{-1} \text{diag}(\mathbf{c}) \\ \Rightarrow \mathbf{\Omega}_{\mathbf{c}}^{-1} &= \text{diag}\left(\frac{1}{\mathbf{c}}\right) \mathbf{U} \text{diag}\left(\frac{1}{\mathbf{c}}\right) \\ &= \mathbf{\Gamma}^{1/2} \mathbf{U} \mathbf{\Gamma}^{1/2} \end{aligned}$$

As $\mathbf{\Gamma}^{1/2}$ is a $D \times D$ diagonal matrix, and that too with sparse diagonal entries. It is easy to see that (i,j)th entry of $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ will be non-zero if both γ_i and γ_j are non zero. For every i for which γ_i is zero, the i^{th} row and the i^{th} column of $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ will be zero. As most of the γ_is are zero, matrix $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ will be sparse. It is easy to see that $\mathbf{\Omega}_{\mathbf{c}}^{-1} \mathbf{A} \mathbf{\Omega}_{\mathbf{c}}^{-1}$ will also consequently be sparse. It will have non-zero entries only at the locations where $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ has non zero entries. The matrix $\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ will also becomes sparse as it is the difference of two sparse matrices having non zero entries at exact same locations.

From Fig. 5a and 5b, we infer that the location of non zero entries in matrix $\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ and $\mathbf{\Omega}_{\mathbf{c}}^{-1}$ are indeed same. Also the matrix $\mathbf{\Omega}_{\mathbf{g}|\mathbf{y}}^{-1}$ is sparse.

B. Proof of Lemma 2

We know that:

$$\mathbf{M} \times \mathbf{M}^{-1} = \mathbf{I}. \tag{14}$$

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & ... & a_{1N} \\ ... & ... & ... & ... \\ a_{N1} & a_{N2} & ... & a_{NN} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & ... & b_{1N} \\ ... & ... & ... & ... \\ b_{N1} & b_{N2} & ... & b_{NN} \end{bmatrix} = \mathbf{I}.$$

Here $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_L$ are the columns of $\mathbf{M}^{-1}$, which are non zero. To retain these columns, we multiply matrix $\mathbf{M}^{-1}$ with matrix $\mathbf{B}$ which is the set of L vectors. These vectors has only one entry equal to 1 and all others 0. For example, suppose 2^{nd}, 3^{rd} and 6^{th} column of matrix $\mathbf{M}^{-1}$ were non-zero, then the required matrix $\mathbf{B} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$. The vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ have 1 at 2^{nd}, 3^{rd} and 6^{th} position respectively and 0 elsewhere. Thus multiplying matrix $\mathbf{B}$ in equation (14) we get:

$$\begin{aligned} &\Rightarrow \mathbf{M} \times \mathbf{M}^{-1} \times \mathbf{B} = \mathbf{I} \times \mathbf{B} = \mathbf{B} \\ &\Rightarrow \mathbf{M} \times \mathbf{X} = \mathbf{B} \\ &\mathbf{X} = linearsolver(\mathbf{M}, \mathbf{B}) \end{aligned}$$

This system of equations can be solved with respect to $\mathbf{X} \in \mathbb{C}^{D \times L}$ in $O(LD^2)$ time. We solve it by using the Conjugate Gradient method [5].

References

[1] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," Journal of Machine Learning Research, vol. 1, pp. 211–244, 2001. [Online]. Available: http://jmlr.csail.mit.edu/papers/v1/tipping01a.html

[2] J. Lee, G.-T. Gil, and Y. H. Lee, "Channel estimation via orthogonal matching pursuit for hybrid mimo systems in millimeter wave communications," IEEE Transactions on Communications, vol. 64, no. 6, pp. 2370–2386, 2016.

[3] D. Prasanna and C. R. Murthy, "mmwave channel estimation via compressive covariance estimation: Role of sparsity and intra-vector correlation," IEEE Transactions on Signal Processing, vol. 69, pp. 2356–2370, 2021.

[4] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," IEEE Transactions on Signal Processing, vol. 61, no. 8, pp. 2009–2015, apr 2013. [Online]. Available: https://doi.org/10.1109%2Ftsp.2013.2241055

[5] A. Lin, A. H. Song, B. Bilgic, and D. Ba, "Covariance-free sparse bayesian learning," 2021. [Online]. Available: https://arxiv.org/abs/2105.10439

[6] C. Bekas, E. Kokiopoulou, and Y. Saad, "An estimator for the diagonal of a matrix," Applied Numerical Mathematics, vol. 57, no. 11, pp. 1214–1229, 2007, numerical Algorithms, Parallelism and Applications (2). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168927407000244