UNIVERSITY OF CALIFORNIA, SAN DIEGO

Sparse Signal Recovery Exploiting Spatiotemporal Correlation

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering (Signal and Image Processing)

by

Zhilin Zhang

Committee in charge:

Professor Bhaskar D. Rao, Chair
Professor Sanjoy Dasgupta
Professor Virginia de Sa
Professor William Hodgkiss
Professor Scott Makeig
Professor Nuno Vasconcelos

2012

The dissertation of Zhilin Zhang is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2012

DEDICATION

To my grandparents, my parents, my wife, and my daughter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Professor Bhaskar D. Rao, who kindly guided my study with patience and encouragement. His keen sense for important research directions stimulated my research. I deeply thank him for bringing me into the exciting field of sparse Bayesian learning, to which I will continue to devote myself after my graduation.

My deep thanks also go to Professor Scott Makeig and Professor Tzyy-Ping Jung in Swartz Center for Computational Neuroscience, who provided me a wonderful environment for study of electrophysiology signals. Without them, I cannot form a deep perspective to the fields of EEG analysis and brain-computer interfaces.

I sincerely thank Professor Li Shen in Indiana University for his valuable input on my research and largely support to my family. Collaborating with him and his group is always a wonderful experience.

I am very grateful for having an exceptional doctoral committee, and thank all my committee members for their valuable suggestions to my work and informative courses. Especially, I wish to thank Professor Nuno Vasconcelos for his course 'Statistical Learning', which helped me build a basis for my study on sparse Bayesian learning.

Special thanks go to Dr. Igor Carron for his kind help. And his *Nuit Blanche*, the most famous blog in the field of compressed sensing, is an important window for me to keep a watch on the progress of this field.

I would like to thank all the colleagues at my DSP Lab, Swartz Center for Computational Neuroscience, and Li Shen Lab in Indiana University for their support, friendship, and assistance in these years.

My study and life was made pleasurable by many excellent friends. Especially, I want to thank Fengyu Cong, Jianjian Gao, Luo Gu, Doug Huntley, Yuzhe Jin, Taiyong Li, Yang Li, Yu Liao, Fei Liu, Wei Lu, Alireza Masnadi-Shirazi, Tim Mullen, Yijun Wang, David Wipf, Wen Qiao, Honghao Shan, Yongxuan

Su, Mingyang Tang, Huajian Yao, Lei Yu, Lelin Zhang, Lingyun Zhang, Xueying Zheng and Yuan Zhi.

I am deeply grateful to my parents for their endless love and unwavering support throughout my long academic journey.

I want to thank my loving wife, Jing, for her love, encouragement, and wisdom. It is the luckiest thing in my life to meet her and marry her.

The text of Chapter II, in full, is a reprint of the material as it appears in: Zhilin Zhang and Bhaskar D. Rao, "Extension of SBL Algorithms for the Recovery of Block Sparse Signals with Intra-Block Correlation", to appear in IEEE Trans. on Signal Processing. The dissertation author was a primary researcher and author of the cited material.

The text of Chapter III, in full, is based on the material as it appears in: Zhilin Zhang and Bhaskar D. Rao, "Sparse Signal Recovery with Temporally Correlated Source Vectors Using Sparse Bayesian Learning", IEEE Journal of Selected Topics in Signal Processing, 2011, Zhilin Zhang and Bhaskar D. Rao, "Iterative Reweighted Algorithms for Sparse Signal Recovery with Temporally Correlated Source Vectors", in Proc. of the 36th International Conference on Acoustics, Speech, and Signal Processing, 2011, and Jing Wan, Zhilin Zhang, Jingwen Yan, Taiyong Li, Bhaskar D. Rao, Shiaofen Fang, Sungeun Kim, Shannon Risacher, Andrew Saykin, Li Shen, "Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer's Disease", in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2012. The dissertation author was a primary researcher and author of the cited papers.

The text of Chapter IV, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Spatiotemporal Sparse Bayesian Learning with Applications to Compressed Sensing of Multichannel ECG for Wireless Telemonitoring", submitted for publication to IEEE Trans. on Biomedical Engineering, 2012, and Zhilin Zhang, Jing Wan, Shiaofen Fang, Andrew Saykin, Li Shen, "Correlation- and Nonlinearity-Aware Sparse Bayesian

Learning with Applications to the Prediction of Cognitive Scores from Neuroimaging Measures", submitted to IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013. The dissertation author was a primary researcher and author of the cited papers.

The text of Chapter V, in part, is currently being prepared for submission for publication of the material: Zhilin Zhang, Bhaskar D. Rao, "Sparse Bayesian Learning for Time-Varying Sparse Model". The dissertation author was a primary researcher and author of this paper.

The text of Chapter VI, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Non-Invasive Fetal ECG via Block Sparse Bayesian Learning", to appear in IEEE Trans. on Biomedical Engineering, 2013. The dissertation author was a primary researcher and author of the cited paper.

The text of Chapter VII, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Spatiotemporal Sparse Bayesian Learning with Applications to Compressed Sensing of Multichannel ECG for Wireless Telemonitoring", submitted to IEEE Trans. on Biomedical Engineering, 2012. The dissertation author was a primary researcher and author of the cited paper.

The text of Chapter VIII, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Compressed Sensing of EEG for Wireless Telemonitoring with Low Energy Consumption and Inexpensive Hardware", to appear in IEEE Trans. on Biomedical Engineering, 2013. The dissertation author was a primary researcher and author of the cited paper.

The text of Chapter IX, in full, is based on the material as it appears in: Zhilin Zhang, Jing Wan, Shiaofen Fang, Andrew Saykin, Li Shen, "Correlation- and Nonlinearity-Aware Sparse Bayesian Learning with Applications to the Prediction

of Cognitive Scores from Neuroimaging Measures", submitted to IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013, and Jing Wan, Zhilin Zhang, Jingwen Yan, Taiyong Li, Bhaskar D. Rao, Shiaofen Fang, Sungeun Kim, Shannon Risacher, Andrew Saykin, Li Shen, "Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer's Disease", in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2012. The dissertation author was a primary researcher and author of the cited papers.

VITA

2002	Bachelor of Science, University of Electronic Science and Technology of China, P.R. China
2005	Master of Science, University of Electronic Science and Technology of China, P.R. China
2005 – 2007	Visiting Scholar, Shanghai Jiao Tong University, P.R. China
2007 – 2012	Research Assistant, Department of Electrical and Computer Engineering, University of California, San Diego, USA
2012	Doctor of Philosophy, University of California, San Diego, USA

PUBLICATIONS

Z.Zhang, T.-P.Jung, S.Makeig, B.D.Rao, "Spatiotemporal Sparse Bayesian Learning with Applications to Compressed Sensing of Multichannel ECG for Wireless Telemonitoring", *IEEE Trans. on Biomedical Engineering (submitted)*, 2012.

Z.Zhang, B.D.Rao, "Extension of SBL Algorithms for the Recovery of Block Sparse Signals with Intra-Block Correlation", *to appear in IEEE Trans. on Signal Processing*

Z.Zhang, T.-P.Jung, S.Makeig, B.D.Rao, "Compressed Sensing of EEG for Wireless Telemonitoring with Low Energy Consumption and Inexpensive Hardware", *to appear in IEEE Trans. on Biomedical Engineering*

Z.Zhang, T.-P.Jung, S.Makeig, B.D.Rao, "Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Noninvasive Fetal ECG via Block Sparse Bayesian Learning", *to appear in IEEE Trans. on Biomedical Engineering*

Z.Zhang, B.D.Rao, "Sparse Signal Recovery with Temporally Correlated Source Vectors Using Sparse Bayesian Learning", *IEEE Journal of Selected Topics in Signal Processing*, vol.5, no.5, pp.912-926, 2011

S.Makeig, C.Kothe, T.Mullen, N.Bigdely-Shamlo, Z.Zhang, K.Kreutz-Delgado, "Evolving Signal Processing for Brain-Computer Interface", *Proceedings of the IEEE*, vol.100, Special Centennial Issue, pp.1567-1584, 2012

B.Liu, Z.Zhang, H.Fan, Z.Lu, Q.Fu, "Fast Marginalized Block SBL Algorithm", *IEEE Signal Processing Letters (submitted)*, 2012.

T.Li, Z.Zhang, "Face Recognition via Block Sparse Bayesian Learning", *Neurocomputing (submitted)*, 2012.

Z.Zhang, J.Wan, S.Fang, A.Saykin, L.Shen, "Correlation- and Nonlinearity-Aware Sparse Bayesian Learning with Applications to the Prediction of Cognitive Scores from Neuroimaging Measures", *submitted to CVPR 2013*

J.Wan*, Z.Zhang*, J.Yan, T.Li, B.D.Rao, S.Fang, S.Kim, S.Risacher, A.Saykin, L.Shen, "Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer's Disease", *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR 2012)*, Rhode Island, USA, June, 2012 (* Equal Contribution)

B.D.Rao, Z.Zhang, Y.Jin, "Sparse Signal Recovery in the Presence of Intra-Vector and Inter-Vector Correlation", *IEEE Int. Conf. on Signal Processing and Communications (SPCOM 2012)*, Bangalore, India, July, 2012

Z.Zhang, B.D.Rao, "Recovery of Block Sparse Signals Using the Framework of Block Sparse Bayesian Learning", *Proc. of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, Japan, March, 2012

Z.Zhang, B.D.Rao, "Iterative Reweighted Algorithms for Sparse Signal Recovery with Temporally Correlated Source Vectors", *Proc. of the 36th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, Prague, the Czech Republic, May, 2011

Z.Zhang, B.D.Rao, "Exploiting Correlation in Sparse Signal Recovery Problems: Multiple Measurement Vectors, Block Sparsity, and Time-Varying Sparsity", *ICML 2011 Workshop on Structured Sparsity: Learning and Inference*, July, 2011

Z.Zhang, B.D.Rao, "Sparse Signal Recovery in the Presence of Correlated Multiple Measurement Vectors", *Proc. of the 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, Texas, USA, 2010

ABSTRACT OF THE DISSERTATION

Sparse Signal Recovery Exploiting Spatiotemporal Correlation

by

Zhilin Zhang
Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)
University of California, San Diego, 2012
Professor Bhaskar D. Rao, Chair

Sparse signal recovery algorithms have significant impact on many fields. The core of these algorithms is to find a solution to an underdetermined inverse system of equations, where the solution is expected to be sparse or approximately sparse. However, most algorithms ignored correlation among nonzero entries of a solution, which is often encountered in a practical problem. Thus, it is unclear what role the correlation plays in signal recovery.

This work aims to design algorithms which can exploit a variety of correlation structures in solutions and reveal the impact of these correlation structures on algorithms' recovery performance.

First, a block sparse Bayesian learning (BSBL) framework is proposed. Based on it, a number of sparse Bayesian learning (SBL) algorithms are derived to exploit intra-block correlation in a block sparse model, temporal correlation in a multiple measurement vector model, spatiotemporal correlation in a spatiotemporal sparse model, and local temporal correlation in a time-varying sparse model. Several optimization approaches are employed in the algorithm development, such as the expectation-maximization method, the bound-optimization method, and a fixed-point method. Experimental results show that these algorithms have superior performance.

With these algorithms, we find that different correlation structures affect the quality of estimated solutions to different degrees. However, if these correlation structures are present and exploited, algorithms' performance can be largely improved. Inspired by this, we connect these algorithms to Group-Lasso type algorithms and iterative reweighted ℓ_1 and ℓ_2 algorithms, and suggest strategies to modify them to exploit the correlation structures for better performance.

The derived algorithms have been used with considerable success in various challenging applications such as wireless telemonitoring of raw physiological signals and prediction of patients' cognitive levels from their neuroimaging measures. In the former application, where raw physiological signals are neither sparse in the time domain nor sparse enough in transformed domains, the derived algorithms are the only algorithms so far that achieved satisfactory results. In the latter application, the derived algorithms achieved the highest prediction accuracy on common datasets, compared to published results around 2011.

Chapter I

Introduction

Sparse signal recovery is a rapidly evolving field having significant impact on many fields, such as signal processing, compressed sensing, information theory, pattern recognition, machine learning, neuroimaging, and bioinformatics. In the following we first introduce some typical mathematical models of sparse signal recovery, and then give some practical scenarios where sparse signal recovery plays an important role.

I.A Models and Algorithms

I.A.1 Single Measurement Vector (SMV) Model

The most basic model in sparse signal recovery is the single measurement vector (SMV) model, given by [56, 57, 129, 21, 16, 15, 37]

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{v}, \tag{I.1}$$

where $\mathbf{y} \in \mathbb{R}^{M\times 1}$ is an available measurement vector, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M \ll N)$ is a known matrix, $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is an unknown vector which we want to estimate, and $\mathbf{v} \in \mathbb{R}^{M\times 1}$ is an unknown noise vector. Generally, the matrix $\mathbf{\Phi}$ is assumed to satisfy the unique representation property (URP) [57], namely any M columns of $\mathbf{\Phi}$ are linearly independent. It has different names in different contexts. For example, in compressed sensing [37], it is called a sensing matrix or a measurement matrix (when $\mathbf{x}$ is a signal to recover). In signal representation, it is called a basis matrix or a dictionary matrix.

The problem (I.1) is an underdetermined inverse problem. Generally, there are infinite solutions. Thus, it is impossible to find the true solution. However, when the true solution is sufficiently sparse (i.e., only a few entries of $\mathbf{x}$ are nonzero), it is possible to find it with small errors [15], or even exactly in some cases [16].

In general, finding the sparsest solution (i.e., a solution with minimal $\|\mathbf{x}\|_0$) to this problem (I.1) requires exhaustive searches over all subsets of columns of $\mathbf{\Phi}$.

However, this approach is NP-hard [97]. Thus, a number of alternative algorithms were proposed to seek the sparsest solution.

One popular family of algorithms are those based on the convex relaxation. With some conditions on $\mathbf{\Phi}$ and $\mathbf{x}$, it can be shown [36, 41, 15] that the true solution of (I.1) can be found within the noise level by solving the following ℓ_1 minimization problem:

$$\begin{aligned} \min_{\mathbf{x}} : \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} : \quad & \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 \leq \delta \end{aligned} \tag{I.2}$$

where δ is a regularizer. Note that there are other equivalent forms, such as the one using the Lagrange multiplier:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{I.3}$$

where λ is another regularizer. There are many solvers to the convex relaxation problems (I.2) or (I.3), such as Lasso [129] and Basis Pursuit Denoising [21]. One drawback of these algorithms is that one needs to tune the regularizer δ or λ. Although there are methods to guide the tuning, such as the L-curve method [59, 60], cross-validation, and model selection [122, 120], in some applications the tuning is very difficult or even impossible. Thus, the selection of optimal λ or δ remains an important topic. Recent progress on this can be found in [130, 126]. Another drawback of these algorithms is that the estimation is generally biased. In other words, the global minimum generally does not correspond to the sparsest solution unless strict conditions on $\mathbf{\Phi}$ and $\mathbf{x}$ are satisfied. When compared to sparse Bayesian learning, they have other drawbacks [155], which will be discussed later.

Non-convex minimization is another family. Algorithms in this family seek a solution with minimal ℓ_p norm, where $0 < p < 1$. Mathematically, they solve the following non-convex minimization problem [1]:

$$\mathbf{x} = \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_p \quad (0 < p < 1) \tag{I.4}$$

[1] For simplicity, we only give the form using the Lagrange multiplier

where $\|\mathbf{x}\|_p$ is defined as

$$\|\mathbf{x}\|_p = \Big(\sum_i |x_i|^p \Big)^{1/p}. \tag{I.5}$$

Solving this problem generally leads to an iterative reweighted algorithm. The most famous one may be the FOCUSS algorithm [57, 108] [2], which has been widely used in Neuromagnetic source localization [55]. These non-convex minimization algorithms also need to solve the issue of optimal choice of λ.

Another family are algorithms based on smooth approximation of ℓ_0 norm of $\mathbf{x}$ [94, 115]. The ℓ_0 norm of $\mathbf{x}$ is defined as $\|\mathbf{x}\|_0 = \sum_i I(x_i \neq 0)$, where $I(\cdot)$ is the indicator function. One advantage of these algorithms is that they have high speed, and have excellent recovery performance in noiseless scenarios. But one drawback is they are not robust to noise. Although some variants can be used for noisy scenarios, their performance is not as impressive as they are in noiseless scenarios.

Another popular family of algorithms are greedy algorithms [87, 99, 134, 27]. Algorithms in this family have high speed. However, their recovery performance is strongly affected by the coherence among columns of $\mathbf{\Phi}$, and also do not have satisfactory performance in noisy scenarios. These drawbacks limit their applications to some problems such as source localization and tracking.

The family of message passing algorithms [38, 141, 10] is a young group, but recently developed algorithms have shown excellent performance in some applications in terms of both speed and recovery performance. However, most algorithms cannot be used in the case when columns of Φ are coherent.

Bayesian algorithms are another powerful algorithms. They can be categorized into two sub-groups. One sub-group are Bayesian counterparts of greedy algorithms, such as the Bayesian pursuit algorithms [113, 162, 64]. Another sub-group are sparse Bayesian learning (SBL) algorithms [132, 153, 127, 105, 70, 49, 7]. SBL has many advantages over other families of algorithms. For example, SBL

[2] However, one should note that FOCUSS can also solve the ℓ_1 minimization problem (I.2) or (I.3).

has good recovery performance in the case when columns of Φ are highly coherent. This property makes it very attractive to the applications such as direction-of-arrive (DOA) estimation, neuroelectromagnetic source localization, earthquake detection, and feature selection in bioinformatics. Compared to other families, another advantage of SBL is that it provides flexibility to model and exploit correlation structures in signals for improved performance [107, 174, 171]. In Section I.C we will discuss its advantages and disadvantages in detail.

I.A.2 Block Sparse Model

In applications, the signal $\mathbf{x}$ generally has additional structures. A widely studied structure is block/group structure [160, 9, 43, 124, 46]. With this structure, $\mathbf{x}$ can be viewed as a concatenation of blocks, i.e.,

$$\mathbf{x} = [\underbrace{x_1, \cdots, x_{d_1}}_{\mathbf{x}_1^T}, \cdots, \underbrace{x_{d_{g-1}+1}, \cdots, x_{d_g}}_{\mathbf{x}_g^T}]^T \tag{I.6}$$

where $d_i(\forall i)$ are not necessarily identical. Among these blocks, only a few blocks are nonzero but their locations are unknown. The SMV model (I.1) with the block partition (I.6) is called *the canonical block sparse model*. It is known that exploiting such block partition can further improve recovery performance.

A number of algorithms have been proposed to recover sparse signals with the block structure. However, few of them consider intra-block correlation, i.e., the correlation among amplitudes of the elements within each block. In practical applications the intra-block correlation widely exists, such as physiological signals and images.

In Chapter II we will derive several algorithms that explore and exploit the intra-block correlation to improve performance. As we will see, the derived algorithms have superior performance to existing algorithms, and has the unique ability to recover non-sparse signals directly.

I.A.3 Multiple Measurement Vector (MMV) Model

The SMV model (I.1) can be used in many applications, such as source localization, radar detection and other DOA estimation scenarios. But in these applications, generally a sequence of measurement vectors are available. Thus the basic SMV model (I.1) can be extended to the following multiple measurement vector (MMV) model [106, 25],

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} + \mathbf{V}, \tag{I.7}$$

where $\mathbf{Y} \triangleq [\mathbf{Y}_{\cdot 1}, \cdots, \mathbf{Y}_{\cdot L}] \in \mathbb{R}^{M\times L}$ consists of L measurement vectors, $\mathbf{X} \triangleq [\mathbf{X}_{\cdot 1}, \cdots, \mathbf{X}_{\cdot L}] \in \mathbb{R}^{N\times L}$ is the desired solution matrix, and $\mathbf{V}$ is an unknown noise matrix. A key assumption in the MMV model is that the support (i.e. indexes of nonzero entries) of every column in $\mathbf{X}$ is identical (referred as *the common sparsity assumption* in literature [25]). In addition, similar to the constraint in the SMV model, the number of nonzero rows in $\mathbf{X}$ has to be below a threshold to ensure a unique and global solution [25]. This leads to the fact that $\mathbf{X}$ has a small number of nonzero rows. It has been shown that compared to the SMV case, the successful recovery rate of the support can be greatly improved using multiple measurement vectors [25, 44, 45, 72].

It is worth pointing out that in practical applications (e.g. source localization), there is correlation among the entries in each nonzero row of $\mathbf{X}$. If ignoring the correlation, it can deteriorates algorithms' recovery performance. Unfortunately, most existing MMV algorithms ignored the correlation. In Chapter III we will introduce several SBL algorithms that can exploit the correlation. We will see by exploiting the temporal correlation we can achieve much better performance than state-of-the-art MMV algorithms.

I.A.4 Time-Varying Sparse Model

Although the MMV model (I.7) is a popular model for source localization, one needs to note that under the common sparsity assumption we cannot obtain

many measurement vectors in practical applications. The main reason is that the sparsity profile of practical signals is (slowly) time-varying, so the common sparsity assumption is valid for only a small L in the MMV model. For example, in EEG/MEG source localization there is considerable evidence [92] that a given pattern of dipole-source distributions [3] may only exist for 10-20 ms. Since the EEG/MEG sampling frequency is generally 250 Hz, a dipole-source pattern may only exist through 5 snapshots (i.e. in the MMV model $L = 5$). In DOA estimation [24], directions of targets [4] are continuously changing, and thus the source vectors that satisfy the common sparsity assumption are few. Of course, one can increase the measurement vector number at the cost of increasing the source number, but a larger source number can result in degraded recovery performance. Thus, the time-varying sparsity model is called for.

The time-varying sparsity model is a natural extension of the MMV model. It considers the case when the support of each column of $\mathbf{X}$ is time-varying. The transition from the stationary models, assumed so far, to the non-stationary situation opens up an abundance of options akin to past work on tracking which has led to adaptive filters, Kalman Filters and so on.

The measurement model in this case is given by

$$\mathbf{y}_t = \mathbf{\Phi}\mathbf{x}_t + \mathbf{v}_t,\ t = 0, 1, 2, ... \tag{I.8}$$

Here, $\mathbf{y}_t \in \mathbb{R}^{M\times 1}$ is a measurement vector, $\mathbf{x}_t \in \mathbb{R}^{N\times 1}$ is the sparse signal with time-varying sparsity, and $\mathbf{v}_t$ is a noise vector.

To deal with time-varying sparsity, many algorithms have been proposed, such as algorithms based on Kalman filters [140] or Bayesian estimation/prediction [161, 114], algorithms based on the sparsity of the unknown part of the support [139], algorithms based on message passing [177], and algorithms based on homotopy continuation principles [109]. Most of the algorithms [140, 114, 139, 161] employ SMV algorithms to find the "turn-on" coefficients at each snapshot.

[3] In this application the set of indexes of nonzero rows in $\mathbf{X}$ is called a pattern of dipole-source distribution.

[4] In this application the index of a nonzero row in $\mathbf{X}$ indicates a direction.

However, since the support of $\mathbf{x}_t$ is changing slowly, we can view such a time-varying sparsity model as a concatenation of several MMV models [163, 170], where in each MMV model the support does not change. Thus, MMV algorithms can be used to solve the time-varying sparse problem. This treatment has several obvious advantages. One advantage is that the support recovery rate is greatly improved because of the enhanced support-recovery ability afforded by the MMV models. The second advantage is that the temporal correlation in each MMV model can be exploited to further improve the support recovery rate. In Chapter V we will introduce an SBL algorithm for the time-varying sparse model.

I.A.5 Spatiotemporal Sparse Model

A spatiotemporal sparse model is another MMV model with different assumptions on $\mathbf{X}$. It can be described as:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} + \mathbf{V}, \tag{I.9}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times L}$, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M < N)$, and $\mathbf{X} \in \mathbb{R}^{N\times L}$. What makes the model different from the previous MMV model is the assumption that the matrix $\mathbf{X}$ has spatiotemporal correlation; the entries in the same nonzero row of $\mathbf{X}$ are correlated, and the nonzero entries in the same column of $\mathbf{X}$ are also correlated. Particularly, we consider the following specific structure of $\mathbf{X}$ in this thesis:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{[1]\cdot} \\ \mathbf{X}_{[2]\cdot} \\ \vdots \\ \mathbf{X}_{[g]\cdot} \end{bmatrix} \tag{I.10}$$

where $\mathbf{X}_{[i]\cdot} \in \mathbb{R}^{d_i\times L}$ is the i-th block of $\mathbf{X}$, and $\sum_{i=1}^{g} d_i = N$. For convenience, $\{d_1, \cdots, d_g\}$ is called the block partition. Among the g blocks, only a few are nonzero blocks. The key assumption is that each block $\mathbf{X}_{[i]\cdot}(\forall i)$ is assumed to have spatiotemporal correlation. In other words, entries in each column of $\mathbf{X}_{[i]\cdot}$ are correlated (intra-block correlation), and entries in each row of $\mathbf{X}_{[i]\cdot}$ are also

correlated (temporal correlation). Thus, the model can be seen as the combination of the canonical MMV model and the canonical block sparse model.

There are few algorithms for this spatiotemporal sparse model. In Chapter IV we will introduce several SBL algorithms for this model, and will show its applications to feature selection and compressed sensing of multichannel physiological signals.

I.B Applications

Sparse signal recovery has been successfully deployed in a variety of applications, including

- EEG/MEG source localization [85, 55, 148]
- adaptive signal processing [6]
- array signal processing [86, 76]
- high-dimensional statistics [47, 91]
- wireless telemonitoring of physiological signals [167, 168, 169, 88]
- wireless sensor networks [62, 39]
- pattern recognition [156]
- rapid MR imaging [82]
- radar imaging [104]
- speech and audio processing [53, 3]
- medical data analysis [143, 125, 164]
- astronomical data analysis [12, 121]
- exploration seismology [63]

- financial data analysis [48]
- neuroscience [52, 69]

The applications introduced below are just a tip of iceberg. For simplicity, only the basic SMV model (I.1) is discussed in the following applications. But one should be aware that many other models can be used in these applications for better performance.

I.B.1 Data Compression

In this application, $\mathbf{x}$ is an original signal, and generally it has sparse representation under some orthonormal bases. That is, $\mathbf{x} = \mathbf{D}\mathbf{z}$, where $\mathbf{D}$ is an orthonormal basis matrix and $\mathbf{z}$ is a sparse vector. The orthonormal basis matrices are often formed from wavelets, noiselets [23], and the discrete cosine transform (DCT). The signal $\mathbf{x}$ is compressed to $\mathbf{y}$ according to the SMV model (I.1), i.e.,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} \tag{I.11}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M < N)$ is a sensing matrix incoherent with $\mathbf{D}$. For most basis matrices, when $\mathbf{\Phi}$ is a random matrix (e.g. a random Gaussian matrix), it is largely incoherent with $\mathbf{D}$.

To recover $\mathbf{x}$, a sparse signal recovery algorithm solves the following underdetermined inverse problem:

$$\mathbf{y} = \mathbf{\Omega}\mathbf{z} \tag{I.12}$$

where $\mathbf{\Omega} \triangleq \mathbf{\Phi}\mathbf{D}$, which is known. Once have recovered the sparse vector $\mathbf{z}$, the original signal $\mathbf{x}$ can be immediately obtained according to $\mathbf{x} = \mathbf{D}\mathbf{z}$.

Compared to traditional data compression approaches, an advantage of sparse signal recovery is that it can be used in the situations when the sparse basis $\mathbf{D}$ is unknown at the encoder or impractical to implement for data compression [167, 168].

For most algorithms, the success of recovery heavily relies on whether the signal $\mathbf{x}$ has sparse representation. Although it was claimed that many signals are sparse under some bases, careful examination is required in some applications. For example, in energy-efficient wireless telemonitoring of a physiological signal, the raw physiological signal is not sparse under various wavelet and DCT bases [167, 168, 169]; the representation coefficient vector $\mathbf{z}$ has a few nonzero coefficients with significant values and a large number of coefficients with small values. Although recovering the coefficients with large values is helpful to restore main characteristics of the original physiological signal $\mathbf{x}$, recovering the coefficients with small values is important to maintain detailed and local characteristics of $\mathbf{x}$. These detailed and local characteristics is often more important and meaningful for clinical diagnosis and other successive signal processing and pattern recognition [90]. In fact, probably in all the data compression applications, recovering both the two kinds of coefficients is always desirable: the more coefficients are recovered, the better the recovery quality is.

However, most algorithms can only recover coefficients with large values. This drawback was recently alleviated with the use of BSBL algorithms and spatiotemporal SBL algorithms, which will be shown in later chapters.

I.B.2 Feature Selection

Sparse signal recovery algorithms have been widely used for feature selection in many applications, such as bioinformatics, financial data analysis, speech processing, and image analysis. Here the application to diagnosis of Alzheimer's disease (AD) is discussed [143, 175, 125].

There is a basic question in the diagnosis of AD: which brain areas determine cognition level of a patient with AD? One can answer the question by setting up an SMV model connecting the cognition levels and the MRI measures of patients as follows:

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{v}$$

where:

- $\mathbf{y} \in \mathbb{R}^{M\times 1}$ are cognitive scores of all the M subjects, which are given by a scoring system when they performed a cognitive task;
- $\mathbf{\Phi}$ is an MRI measure matrix of all the subjects. Each column consists of the MRI measures on a brain area of all the subjects. Particularly, $\Phi_{j,k}$ is the MRI measure of the k-th brain area of the j-th subject.
- $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is the regression coefficient vector. A significantly nonzero entry of $\mathbf{x}$, say x_q, means that the MRI measure of the q-th brain area have strong influence on the cognitive scores of all subjects.

The coefficient vector $\mathbf{x}$ is expected to be sparse, since the brain circuitry relevant to a certain cognition task typically involves a small number of brain areas, and the MRI measures of these brain areas more or less affect all the cognitive scores under the task.

Thus, one can use sparse signal recovery algorithms to estimate $\mathbf{x}$, and the estimated nonzero entries indicate which brain areas are related to the cognition levels of patients with AD.

I.C Why Choose Sparse Bayesian Learning

Sparse Bayesian learning (SBL) is a powerful Bayesian variable selection methodology, especially when the number of useful variables is small. It was first proposed by Tipping [132, 131, 133]. Then it drew much attention in machine learning [49], mainly viewed as Bayesian support vector machine. Later, it was introduced to the field of sparse signal recovery by Wipf and Rao [153] as a method of basis selection for sparse linear regression models. A number of theoretical results [151, 149, 155, 150] have been obtained by them, showing its advantages over many popular algorithms. On the other hand, many SBL variants have been derived for sparse signal recovery and compressed sensing. While most algorithms

were derived for the basic SMV model [153, 70, 105, 127, 7, 118, 158], there are several algorithms derived for the MMV model [172, 173, 170, 174, 143, 154], for the block sparse model [171, 165, 167, 168, 159, 79], for the time-varying sparse model [107, 170], and for the spatiotemporal sparse model [175, 169].

SBL has drawn much attention in sparse signal recovery and compressed sensing due to a number of advantages over other algorithms:

1. It provides large flexibility to model and exploit correlation structure in signals, such as temporal correlation [172, 170, 173, 174, 143], intra-block correlation [171, 165, 167, 168, 79], and spatiotemporal correlation [175, 169]. By exploiting the correlation structures, recovery performance is significantly improved. Since natural signals (e.g. physiological signals, images, speech signals, and seismic waves) have always correlation structures, it is not surprising that SBL algorithms achieved top performance in many practical problems, or even solved some bottlenecks which other sparse signal recovery algorithms cannot solve [167, 168]. Besides, it is interesting to see that SBL has connections to Lasso-type algortihms [171, 143, 173, 170], therefore, one can modify existing Lasso-type algorithms or design new Lasso-type algorithms to exploit the correlation structures for better performance.

2. Its recovery performance is robust to the characteristics of the matrix $\mathbf{\Phi}$, while other algorithms are not. For example, it has been shown that when columns of $\mathbf{\Phi}$ are highly coherent, SBL still maintains good performance, while other algorithms such as Lasso or other algorithms based on convex relaxation have seriously degraded performance [150]. Experiments also showed that when $\mathbf{\Phi}$ is a non-random matrix or a sparse matrix, SBL algorithms maintain excellent performance, while some algorithms such as some message passing algorithms have poor performance. This advantage is very attractive to feature selection in bioinformatics, source localization, and other applications, since in these applications the matrix $\mathbf{\Phi}$ is not a random matrix and its columns are highly correlated.

3. The recently proposed block sparse Bayesian learning (BSBL) framework [174, 171] has the ability to find non-sparse true solutions to underdetermined inverse problems with sufficiently small errors, as long as the entries in solution vectors or solution matrices are correlated [167, 168, 169]. This is a desired ability, since practical signals are not strictly sparse, and their representation under some dictionary matrices (e.g. the orthonormal basis of wavelets) may not be sufficiently sparse. For example, in [167, 168, 169] it is empirically shown that electroencephalogram signals and electrocardiogram signals do not have strictly sparse representation under the orthonormal basis of popular wavelets and discrete cosine transform. Recovering non-sparse signals is very important to improve the recovery quality of practical signals. It is worth pointing out that so far there are no other algorithms with the same ability.

4. SBL has a number of desired advantages over many popular algorithms in terms of local and global convergence. It can be shown that SBL provides a sparser solution than Lasso-type algorithms [149]. In particular, in noiseless situations and under certain conditions, the global minimum of SBL cost function is unique and corresponds to the true sparsest solution, while the global minimum of the cost function of Lasso-type algorithms is not necessarily the true sparsest solution [153, 154]. Besides, it can be shown [155] that in certain settings, Lasso-type algorithms and $\ell_p (p < 1)$ minimization algorithms always fail, while SBL successes, regardless of $\mathbf{\Phi}$ and sparsity of $\mathbf{x}$. These advantages imply that SBL is a better choice in feature selection [143], EEG/MEG source localization [152, 85], and so on.

5. SBL provides scale-invariant solutions, while Lasso-type algorithms cannot [155]. Let $\mathbf{x}_{\mathrm{SBL}}$ be the optimal solution provided by SBL with the sensing matrix $\mathbf{\Phi}$. A scale-invariant solution means that when rescaling $\mathbf{\Phi}$ with a diagonal matrix $\mathbf{D}$, i.e., $\mathbf{\Phi} \rightarrow \mathbf{\Phi D}$, the optimal solution provided by SBL

becomes $\mathbf{D}\mathbf{x}_{\mathrm{SBL}}$. In contrast, for Lasso-type algorithms, there is no such linear relationship between the solutions. For example, the solution to the problem $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ has no such linear relationship with the solution to the rescaled problem $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$. This analysis warns that rescaling $\mathbf{\Phi}$ may be problematic in EEG/MEG source localization and other regression applications when using Lasso-type algorithms.

Admittedly, SBL is not perfect. The main drawback is that SBL generally involves large computational load. Some strategies have been used to speed up SBL. For example, using the marginalized likelihood method [133], several fast algorithms have been derived [79, 70]. Using the connection between SBL and Lasso-type algorithms [149, 171, 143, 170], one can obtain optimal SBL solutions by iteratively performing Lasso-type algorithms several times. Since Lasso-type algorithms become more efficient year by year, using this iteration strategy also greatly benefits SBL. However, SBL is still slower than some efficient algorithms, such as greedy algorithms or message passing algorithms. Thus, new strategies are desired to speed up SBL, and more efficient SBL algorithms are needed.

Another drawback of SBL is that the estimation of noise variance is not reliable. Learning rules for the noise variance in most SBL algorithms are not effective in noisy environments. Thus, most SBL algorithms [70, 153, 154, 152] use some fixed sub-optimal values, or require users or other algorithms to provide the value, instead of learning it. Recently, an effective empirical strategy to enhance these learning rules has been proposed [174, 171], which helps SBL achieve satisfactory solutions. However, this strategy does not completely solve this problem. More effective methods and theoretical guidance are called for.

I.D Contributions

The contributions of this work can be summarized as follows:

1. From the perspective of methodologies, the work explores and exploits corre-

lation structures in sparse signal recovery models. In particular, algorithms to explore and exploit intra-block correlation in the block sparse model [171, 165], temporal correlation in the canonical MMV model [174, 172, 173, 143], spatiotemporal correlation in the spatiotemporal sparse model [169, 175], and temporal correlation in short durations in the time-varying sparse model have been developed. It has been shown that exploiting the correlation structures in addition to sparsity can significantly improve recovery performance, and it is crucial to the recovery of natural signals which are not sparse or have no sparse representations [167, 168, 169].

2. From the perspective of algorithm frameworks, this work proposes the block sparse Bayesian learning (BSBL) framework [171, 174]. The BSBL framework is the key to solving inverse problems in the block sparse model, the MMV model, the spatiotemporal sparse model and the time-varying sparsity model. It also provides flexibility to exploit various kinds of correlation structures in these models. More importantly, it has the ability to recover signals which are not sparse or have no sparse representations [167, 168, 169]. So far, only the BSBL framework has such ability in the field.

3. From the perspective of algorithms,

   - SBL algorithms have been developed for the MMV model [174, 172, 173, 143], the block sparse model [171, 165, 79], the spatiotemporal sparse model [169, 175], and the time-varying sparse model. Most of them have the best recovery performance among existing algorithms in the field. These algorithms largely enrich the SBL family.
   - By connecting the derived SBL algorithms to iterative reweighted ℓ_1 algorithms [143, 170] and iterative reweighted ℓ_2 algorithms [173], a new family of sparse penalties which exploit correlation have been derived. By combining these penalties and traditional sparsity-encouraging penalties, one can design new algorithms according to specific tasks or re-

quirements.

- Motivated by the connection between Group-Lasso-type algorithms and BSBL algorithms, strategies [171, 143, 173, 170] to modify existing Group-Lasso-type algorithms and mixed-norm minimization algorithms to explore and exploit correlation structures have been proposed. With these modifications, these existing algorithms have significant improved performance.
- Automatically estimating the regularizer λ (which is modeled as noise variance) in most SBL algorithms is problematic in practical noisy environments. By modifying some quantities in the estimation procedures, this problem has been solved to a large degree [174, 171]. With this modification, existing SBL algorithms have improved performance and do not need users to assign fixed values, which largely relax users' burden in the tasks of feature selection, source localization and so on.

4. From the perspective of applications, the derived algorithms have achieved remarkable success in various applications, including compressed sensing [167, 168, 169], feature selection [143, 175], face recognition [78], wireless communication, EEG/MEG source localization, brain connectivity analysis, and earthquake detection, especially in the following two applications.

    - The derived BSBL algorithms and spatiotemporal SBL algorithms solved the bottleneck in compressed sensing of raw ECG and EEG recordings for energy-efficient wireless telemonitoring [167, 168, 169]. So far, no other compressed sensing algorithm can solve this bottleneck, since these raw recordings are not sparse in the time domain and also not sufficiently sparse in transformed domains (e.g. the wavelet domain and the DCT domain)
    - The derived T-MSBL algorithms and spatiotemporal SBL algorithms achieved much higher accuracy in predicting patients' cognition lev-

els from their MRI measures than traditional and state-of-the-art algorithms [143, 175].

Chapter II

Sparse Bayesian Learning Exploiting Intra-Block Correlation

In Chapter I.A.2 we have introduced the canonical block sparse model. Its mathematical expression is given by

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{v}, \tag{II.1}$$

where $\mathbf{y} \in \mathbb{R}^{M\times 1}$, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M \ll N)$, $\mathbf{x} \in \mathbb{R}^{N\times 1}$, and $\mathbf{v} \in \mathbb{R}^{M\times 1}$. $\mathbf{x}$ has the block structure, i.e.,

$$\mathbf{x} = [\underbrace{x_1, \cdots, x_{d_1}}_{\mathbf{x}_1^T}, \cdots, \underbrace{x_{d_{g-1}+1}, \cdots, x_{d_g}}_{\mathbf{x}_g^T}]^T \tag{II.2}$$

where $d_i(\forall i)$ are not necessarily identical. Among the g blocks, only k blocks are nonzero, where $k \ll g$. A signal with the block structure is called a *block sparse signal*. For a block sparse signal, exploiting its block structure by employing the block sparse model can achieve better performance than employing the basic SMV model [124, 9, 43].

There are many algorithms for this model. Typical algorithms include Model-CoSaMp [9], Block-OMP [43], and Group-Lasso type algorithms such as the original Group Lasso algorithm [160], Group Basis Pursuit [137], and Mixed ℓ_2/ℓ_1 Program [44]. These algorithms require to know the block partition. Other algorithms do not need to know the block partition, but need to know other a priori information (e.g. the number of nonzero elements), such as StructOMP [65]. Very recently, CluSS-MCMC [159] and BM-MAP-OMP [101] are proposed, which require very little a priori knowledge.

However, few of existing algorithms consider intra-block correlation, i.e., correlation among amplitudes of elements within each block. But in fact, the intra-block correlation widely exists in practical signals, such as physiological signals [167], images, and wavelet coefficients of speech. For example, in the compressed sensing of an image, each block corresponds to a patch in the image, while in each individual patch, pixels have very similar tone indicating their amplitudes are highly correlated (if the image is modeled as a random field).

This chapter derives a number of algorithms that adaptively learn and exploit the intra-block correlation for better recovery performance. The proposed

algorithms are the first ones in the category that *adaptively* learn and exploit the intra-block correlation for improved recovery performance. Extensive simulations and experiments on real-world datasets are conducted, showing that the algorithms significantly outperform competitive algorithms especially when such correlation is high.

By connecting these algorithms to the Group-Lasso type algorithms, a promising strategy is proposed to incorporate the intra-block correlation in the Group-Lasso type algorithms to improve their performance.

Insight into the effect of the intra-block correlation on algorithms' performance is also given. It is generally viewed that an MMV model is a special case of a block sparse model. But we found the effect of the intra-block correlation on algorithms' performance is quite different from the effect of the temporal correlation in an MMV model [174].

For the situation when the block partition (II.2) is unknown, a simple approximate model and associated algorithms are proposed. These algorithms are effective especially in noisy environments.

In this chapter bold symbols are reserved for vectors and matrices. For square matrices $\mathbf{A}_1, \cdots, \mathbf{A}_g$, $\mathrm{diag}\{\mathbf{A}_1, \cdots, \mathbf{A}_g\}$ denotes a block diagonal matrix with principal diagonal blocks being $\mathbf{A}_1, \cdots, \mathbf{A}_g$ in turn. $\mathrm{Tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$. $\boldsymbol{\gamma} \succeq \mathbf{0}$ means each element in the vector $\boldsymbol{\gamma}$ is nonnegative.

II.A Block Sparse Bayesian Learning Framework

For the block sparse model, we proposed a block sparse Bayesian learning (BSBL) framework [165, 171]. It is an extension of the basic sparse Bayesian learning (SBL) framework [132, 153]. In this framework, each block $\mathbf{x}_i \in \mathbb{R}^{d_i \times 1}$ is assumed to satisfy a parameterized multivariate Gaussian distribution:

$$p(\mathbf{x}_i; \gamma_i, \mathbf{B}_i) \sim \mathcal{N}(\mathbf{0}, \gamma_i \mathbf{B}_i) \quad i = 1, \cdots, g \tag{II.3}$$

with the unknown hyperparameters γ_i and $\mathbf{B}_i$. Here γ_i is a nonnegative parameter controlling the block-sparsity of $\mathbf{x}$. When $\gamma_i = 0$, the i-th block becomes zero. During the learning procedure most γ_i tend to be zero, due to the mechanism of automatic relevance determination [98]. Thus sparsity in the block level is encouraged. $\mathbf{B}_i \in \mathbb{R}^{d_i \times d_i}$ is a positive definite matrix, capturing the correlation structure of the i-th block. Under the assumption that blocks are mutually uncorrelated, the prior of $\mathbf{x}$ is

$$p(\mathbf{x}; \{\gamma_i, \mathbf{B}_i\}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_0) \tag{II.4}$$

where $\mathbf{\Sigma}_0$ is a block-diagonal matrix with the i-th principal block given by $\gamma_i \mathbf{B}_i$, i.e.,

$$\mathbf{\Sigma_0} \triangleq \begin{bmatrix} \gamma_1 \mathbf{B}_1 & & & \\ & \gamma_2 \mathbf{B}_2 & & \\ & & \ddots & \\ & & & \gamma_g \mathbf{B}_g \end{bmatrix}. \tag{II.5}$$

Besides, assume the noise vector has the parameterized multivariate Gaussian distribution

$$p(\mathbf{v}; \lambda) \sim \mathcal{N}(\mathbf{0}, \lambda \mathbf{I}) \tag{II.6}$$

where λ is a positive scalar. Therefore the posterior of $\mathbf{x}$ is given by

$$p(\mathbf{x}|\mathbf{y}; \lambda, \{\gamma_i, \mathbf{B}_i\}_{i=1}^g) = \mathcal{N}(\boldsymbol{\mu}_x, \mathbf{\Sigma}_x) \tag{II.7}$$

with

$$\boldsymbol{\mu}_x = \mathbf{\Sigma}_0 \mathbf{\Phi}^T \left(\lambda \mathbf{I} + \mathbf{\Phi} \mathbf{\Sigma}_0 \mathbf{\Phi}^T\right)^{-1} \mathbf{y} \tag{II.8}$$

$$\mathbf{\Sigma}_x = (\mathbf{\Sigma}_0^{-1} + \frac{1}{\lambda} \mathbf{\Phi}^T \mathbf{\Phi})^{-1} \tag{II.9}$$

Once the parameters $\lambda, \{\gamma_i, \mathbf{B}_i\}_{i=1}^g$ are estimated, the Maximum-A-Posterior (MAP) estimate of $\mathbf{x}$, denoted by $\widehat{\mathbf{x}}$, can be directly obtained from the mean of the posterior, i.e. $\widehat{\mathbf{x}} = \boldsymbol{\mu}_x$.

The parameters are generally estimated by the Type II maximum likelihood procedure [132, 84]. This is equivalent to minimize the following negative log-likelihood with respect to each parameter

$$\begin{aligned}\mathcal{L}(\Theta) &\triangleq -2\log\int p(\mathbf{y}|\mathbf{x};\lambda)p(\mathbf{x};\{\gamma_i,\mathbf{B}_i\}_i)d\mathbf{x} \\ &= \log|\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T|+\mathbf{y}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T)^{-1}\mathbf{y}, \qquad \text{(II.10)}\end{aligned}$$

where Θ denotes all the parameters, i.e., $\Theta \triangleq \{\lambda, \{\gamma_i, \mathbf{B}_i\}_{i=1}^g\}$. This framework is called the BSBL framework [165, 171].

To derive learning rules for these parameters, one can use various optimization methods. Different optimization methods result in different SBL algorithms. Each algorithm includes three learning rules, i.e., the learning rules for γ_i, $\mathbf{B}_i$, and λ.

The learning rule for γ_i is the main body of an algorithm. Different γ_i learning rules result in different speed [1], and determine the possible best recovery performance when optimal values of λ and $\mathbf{B}_i$ are given.

The λ learning rule is also important. If one is unable to find an optimal (or a good suboptimal) value for λ, the recovery performance can be very poor even if the γ_i learning rule could potentially lead to perfect recovery performance.

As for $\mathbf{B}_i(\forall i)$, it can be shown [174] that in noiseless environments the unique and global minimum of (II.10) always leads to the true sparse solution regardless of the value of $\mathbf{B}_i$. In other words, $\mathbf{B}_i$ only affects the probability to converge to local minima. Therefore, one can impose various constraints on the form of $\mathbf{B}_i$ to achieve better recovery performance and also prevent overfitting.

II.B Algorithms for the Situation When the Block Partition is Known

This section presents three algorithms derived from the BSBL framework. They need to know the block partition. The three algorithms have different char-

[1] The λ learning rule also affects the speed, but its effect is not dominant.

acteristics, which will be discussed in Section II.B.4.

II.B.1 BSBL-EM: Use the EM Method

We first use the Expectation-Maximization (EM) method to derive the learning rules for the parameters. Treating $\mathbf{x}$ as hidden variables, we construct the Q-function

$$\begin{aligned} Q(\Theta) &= E_{\mathbf{x}|\mathbf{y};\Theta^{(\text{old})}}\big[\log p(\mathbf{y},\mathbf{x};\Theta)\big] \\ &= E_{\mathbf{x}|\mathbf{y};\Theta^{(\text{old})}}[\log p(\mathbf{y}|\mathbf{x};\lambda)] + E_{\mathbf{x}|\mathbf{y};\Theta^{(\text{old})}}\big[\log p(\mathbf{x};\{\gamma_i,\mathbf{B}_i\}_i)\big]. \end{aligned}$$

Computing the derivatives of $Q(\Theta)$ w.r.t. γ_i and λ and then setting them to zero, we obtain the learning rules:

$$\gamma_i \leftarrow \frac{1}{d_i}\text{Tr}\big[\mathbf{B}_i^{-1}\big(\boldsymbol{\Sigma}_x^i + \boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T\big)\big], \quad \forall i \tag{II.11}$$

$$\lambda \leftarrow \frac{\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\mu}_x\|_2^2 + \text{Tr}(\boldsymbol{\Sigma}_x\boldsymbol{\Phi}^T\boldsymbol{\Phi})}{M}. \tag{II.12}$$

where $\boldsymbol{\mu}_x^i \in \mathbb{R}^{d_i\times 1}$ is the corresponding i-th block in $\boldsymbol{\mu}_x$, and $\boldsymbol{\Sigma}_x^i \in \mathbb{R}^{d_i\times d_i}$ is the corresponding i-th principal diagonal block in $\boldsymbol{\Sigma}_x$. Note that the λ learning rule (II.12) is not robust in low SNR cases. By numerical study, we empirically find that one of the reasons is the disturbance caused by the off-block-diagonal elements in $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$. Therefore, we set their off-block-diagonal elements to zero, leading to the learning rule

$$\lambda \leftarrow \frac{\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\mu}_x\|_2^2 + \sum_{i=1}^{g}\text{Tr}(\boldsymbol{\Sigma}_x^i(\boldsymbol{\Phi}^i)^T\boldsymbol{\Phi}^i)}{M}, \tag{II.13}$$

where $\boldsymbol{\Phi}^i \in \mathbb{R}^{M\times d_i}$ is the submatrix of $\boldsymbol{\Phi}$, which corresponds to the i-th block of $\mathbf{x}$. This λ learning rule is better than (II.12) in generally noisy environments (e.g., SNR $<$ 20dB). In noiseless cases there is no need to use any λ learning rules. Just fixing λ to a small value, e.g., 10^{-10}, can obtain satisfactory performance.

Computing the derivatives of $Q(\Theta)$ w.r.t. $\mathbf{B}_i$ and setting it to zero, we can also derive a learning rule for $\mathbf{B}_i$. However, assigning each block with a different $\mathbf{B}_i$ can result in overfitting. When blocks have the same size, an effective strategy

to avoid overfitting is parameter averaging, i.e. constraining $\mathbf{B}_i = \mathbf{B}(\forall i)$. Using this constraint, the learning rule for $\mathbf{B}$ can be derived as follows:

$$\mathbf{B} \leftarrow \frac{1}{g}\sum_{i=1}^{g}\frac{\boldsymbol{\Sigma}_x^i + \boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T}{\gamma_i}. \tag{II.14}$$

However, the resulting algorithm's performance can be improved by further constraining the matrix $\mathbf{B}$. The idea is to find a positive definite and symmetric matrix $\widehat{\mathbf{B}}$ so that it is determined by one parameter but is close to $\mathbf{B}$ especially along the main diagonal and the main sub-diagonal. Further, we find that for many applications modeling elements of a block as a first-order Auto-Regressive (AR) process is sufficient to model the intra-block correlation [172]. In this case, the corresponding covariance matrix of the block is a Toeplitz matrix with the following form:

$$\text{Toeplitz}([1, r, \cdots, r^{d-1}]) = \begin{bmatrix} 1 & r & \cdots & r^{d-1} \\ \vdots & & & \vdots \\ r^{d-1} & r^{d-2} & \cdots & 1 \end{bmatrix} \tag{II.15}$$

where r is the AR coefficient and d is the block size. Here we constrain $\widehat{\mathbf{B}}$ to have this form. Instead of estimating r from the BSBL cost function, we empirically calculate its value by $r \triangleq \frac{m_1}{m_0}$, where m_0 (res. m_1) is the average of the elements along the main diagonal (res. the main sub-diagonal) of the matrix $\mathbf{B}$ in (II.14).

When blocks have different sizes, the above idea can still be used. First, using the EM method we can derive the rule for each $\mathbf{B}_i$:

$$\mathbf{B}_i \leftarrow \frac{1}{\gamma_i}\big[\boldsymbol{\Sigma}_x^i + \boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T\big]. \tag{II.16}$$

Then, for each $\mathbf{B}_i$ we calculate the averages of the elements along the main diagonal and the main sub-diagonal, i.e. m_0^i and m_1^i, respectively, and average m_0^i and m_1^i for all blocks as follows: $\overline{m}_0 \triangleq \sum_{i=1}^{g} m_0^i$ and $\overline{m}_1 \triangleq \sum_{i=1}^{g} m_1^i$. Finally, we have $\overline{r} \triangleq \frac{\overline{m}_1}{\overline{m}_0}$, from which we construct $\widehat{\mathbf{B}}_i$ for the i-th block:

$$\widehat{\mathbf{B}}_i = \text{Toeplitz}([1, \overline{r}, \cdots, \overline{r}^{d_i-1}]) \quad \forall i \tag{II.17}$$

We denote the above algorithm by **BSBL-EM**.

II.B.2 BSBL-BO: the Bound-Optimization Method

The derived BSBL-EM has good recovery performance but has slow speed. This is mainly due to the EM based γ_i learning rule. For the basic SBL algorithm, Tipping [132] derived a fixed-point based γ_i learning rule to replace the EM based one, which has faster speed but is not robust in some noisy environments. Here we derive a much fast γ_i learning rule, which is based on the bound-optimization method (also known as the Majorization-Minimization method) [42, 123]. The algorithm adopting this γ_i learning rule is denoted by **BSBL-BO** (it uses the same learning rules for $\mathbf{B}_i$ and λ in BSBL-EM). It not only has fast speed, but also has good performance in noisy environments.

Note that the original cost function (II.10) consists of two terms. The first term $\log|\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T|$ is concave with respect to $\boldsymbol{\gamma}\succeq\mathbf{0}$, where $\boldsymbol{\gamma}\triangleq[\gamma_1,\cdots,\gamma_g]^T$. The second term $\mathbf{y}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T)^{-1}\mathbf{y}$ is convex with respect to $\boldsymbol{\gamma}\succeq\mathbf{0}$. Since our goal is to minimize the cost function, we choose to find an upper-bound for the first item and then minimize the upper-bounded cost function.

We use the supporting hyperplane of the first term as its upper-bound. Let $\boldsymbol{\gamma}^*$ be a given point in the $\boldsymbol{\gamma}$-space. We have

$$\begin{aligned}\log|\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T| &\leq \log|\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0^*\mathbf{\Phi}^T|+\sum_{i=1}^{g}\mathrm{Tr}\big((\mathbf{\Sigma}_y^*)^{-1}\mathbf{\Phi}^i\mathbf{B}_i(\mathbf{\Phi}^i)^T\big)(\gamma_i-\gamma_i^*)\\ &= \sum_{i=1}^{g}\mathrm{Tr}\big((\mathbf{\Sigma}_y^*)^{-1}\mathbf{\Phi}^i\mathbf{B}_i(\mathbf{\Phi}^i)^T\big)\gamma_i+\log|\mathbf{\Sigma}_y^*|\\ &\quad -\sum_{i=1}^{g}\mathrm{Tr}\big((\mathbf{\Sigma}_y^*)^{-1}\mathbf{\Phi}^i\mathbf{B}_i(\mathbf{\Phi}^i)^T\big)\gamma_i^* \qquad \text{(II.18)}\end{aligned}$$

where $\mathbf{\Sigma}_y^*=\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0^*\mathbf{\Phi}^T$ and $\mathbf{\Sigma}_0^*=\mathbf{\Sigma}_0|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^*}$. Substituting (II.18) into the cost function (II.10) we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}) &\leq \sum_{i=1}^{g}\mathrm{Tr}\big((\mathbf{\Sigma}_y^*)^{-1}\mathbf{\Phi}^i\mathbf{B}_i(\mathbf{\Phi}^i)^T\big)\gamma_i+\mathbf{y}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Sigma}_0\mathbf{\Phi}^T)^{-1}\mathbf{y}\\ &\quad +\log|\mathbf{\Sigma}_y^*|-\sum_{i=1}^{g}\mathrm{Tr}\big((\mathbf{\Sigma}_y^*)^{-1}\mathbf{\Phi}^i\mathbf{B}_i(\mathbf{\Phi}^i)^T\big)\gamma_i^*\\ &\triangleq \widetilde{\mathcal{L}}(\boldsymbol{\gamma}) \qquad \text{(II.19)}\end{aligned}$$

The function $\widetilde{\mathcal{L}}(\boldsymbol{\gamma})$ is convex over $\boldsymbol{\gamma}$, and when $\boldsymbol{\gamma} = \boldsymbol{\gamma}^*$, we have $\mathcal{L}(\boldsymbol{\gamma}^*) = \widetilde{\mathcal{L}}(\boldsymbol{\gamma}^*)$. Further, for any $\boldsymbol{\gamma}_{\min}$ which is the minimum point of $\widetilde{\mathcal{L}}(\boldsymbol{\gamma})$, we have the following relationship: $\mathcal{L}(\boldsymbol{\gamma}_{\min}) \leq \widetilde{\mathcal{L}}(\boldsymbol{\gamma}_{\min}) \leq \widetilde{\mathcal{L}}(\boldsymbol{\gamma}^*) = \mathcal{L}(\boldsymbol{\gamma}^*)$. This indicates that when we minimize the surrogate function $\widetilde{\mathcal{L}}(\boldsymbol{\gamma})$ over $\boldsymbol{\gamma}$, the resulting minimum point effectively decreases the original cost function $\mathcal{L}(\boldsymbol{\gamma})$. We can use any convex optimization software to optimize the function (II.19). However, this takes more time than BSBL-EM and experiments have shown that also leads to poorer recovery performance. Therefore, we consider another surrogate function. Using the identity:

$$\mathbf{y}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T)^{-1}\mathbf{y} \equiv \min_{\mathbf{x}} \big[\frac{1}{\lambda}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 + \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x}\big] \tag{II.20}$$

where the optimal $\mathbf{x}$ is $\boldsymbol{\mu}_x$, we have

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\gamma}) \;&=\; \min_{\mathbf{x}} \frac{1}{\lambda}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 + \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \sum_{i=1}^{g} \mathrm{Tr}\big((\boldsymbol{\Sigma}_y^*)^{-1}\boldsymbol{\Phi}^i\mathbf{B}_i(\boldsymbol{\Phi}^i)^T\big)\gamma_i \\ &\quad + \log|\boldsymbol{\Sigma}_y^*| - \sum_{i=1}^{g} \mathrm{Tr}\big((\boldsymbol{\Sigma}_y^*)^{-1}\boldsymbol{\Phi}^i\mathbf{B}_i(\boldsymbol{\Phi}^i)^T\big)\gamma_i^*. \end{aligned} \tag{II.21}$$

Then, a new function

$$\begin{aligned}\mathcal{G}(\boldsymbol{\gamma}, \mathbf{x}) \;&\triangleq\; \frac{1}{\lambda}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 + \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \sum_{i=1}^{g} \mathrm{Tr}\big((\boldsymbol{\Sigma}_y^*)^{-1}\boldsymbol{\Phi}^i\mathbf{B}_i(\boldsymbol{\Phi}^i)^T\big)\gamma_i \\ &\quad + \log|\boldsymbol{\Sigma}_y^*| - \sum_{i=1}^{g} \mathrm{Tr}\big((\boldsymbol{\Sigma}_y^*)^{-1}\boldsymbol{\Phi}^i\mathbf{B}_i(\boldsymbol{\Phi}^i)^T\big)\gamma_i^* \end{aligned} \tag{II.22}$$

is defined, which is the upper-bound of $\widetilde{\mathcal{L}}(\boldsymbol{\gamma})$. Note that $\mathcal{G}(\boldsymbol{\gamma}, \mathbf{x})$ is convex in both $\boldsymbol{\gamma}$ and $\mathbf{x}$. It can be easily shown that the solution $(\boldsymbol{\gamma}^\diamond)$ of $\widetilde{\mathcal{L}}(\boldsymbol{\gamma})$ is the solution $(\boldsymbol{\gamma}^\diamond, \mathbf{x}^\diamond)$ of $\mathcal{G}(\boldsymbol{\gamma}, \mathbf{x})$. Thus, $\mathcal{G}(\boldsymbol{\gamma}, \mathbf{x})$ is our final surrogate cost function.

Taking the derivative of $\mathcal{G}$ with respect to γ_i, we can obtain

$$\gamma_i \;\leftarrow\; \sqrt{\frac{\mathbf{x}_i^T\mathbf{B}_i^{-1}\mathbf{x}_i}{\mathrm{Tr}\big((\boldsymbol{\Phi}^i)^T(\boldsymbol{\Sigma}_y^*)^{-1}\boldsymbol{\Phi}^i\mathbf{B}_i\big)}} \tag{II.23}$$

Due to this γ_i learning rule, BSBL-BO takes much fewer iterations than BSBL-EM, but has almost the same recovery performance as BSBL-EM.

II.B.3 BSBL-ℓ_1: the Hybrid of BSBL and Group-Lasso Type Algorithms

Since the cost function of BSBL-EM and BSBL-BO is a function of $\boldsymbol{\gamma}$, they essentially operate in the $\boldsymbol{\gamma}$-space. In contrast, most existing algorithms for the block sparse model directly operate in the $\mathbf{x}$-space, minimizing a data fit term and a penalty which are both functions of $\mathbf{x}$. It is interesting to see the relation between our BSBL algorithms and those algorithms.

Using the idea we presented in [170], an extension of the duality space analysis for the basic SBL framework [149], we can transform the BSBL cost function (II.10) from the $\boldsymbol{\gamma}$-space to the $\mathbf{x}$-space. Since λ and $\mathbf{B}_i(\forall i)$ are regularizers, for convenience we first treat them as fixed values.

First, using the identity (II.20) we can upper-bound the BSBL cost function as follows:

$$\mathfrak{L}(\mathbf{x},\boldsymbol{\gamma}) = \log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T| + \frac{1}{\lambda}\|\mathbf{y}-\boldsymbol{\Phi}\mathbf{x}\|_2^2 + \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x}. \tag{II.24}$$

By first minimizing over $\boldsymbol{\gamma}$ and then minimizing over $\mathbf{x}$, we have:

$$\mathbf{x} = \arg\min_{\mathbf{x}}\Big\{\|\mathbf{y}-\boldsymbol{\Phi}\mathbf{x}\|_2^2 + \lambda g_{\mathrm{c}}(\mathbf{x})\Big\}, \tag{II.25}$$

with the penalty $g_{\mathrm{c}}(\mathbf{x})$ given by

$$g_{\mathrm{c}}(\mathbf{x}) \triangleq \min_{\boldsymbol{\gamma}\succeq\mathbf{0}}\Big\{\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T|\Big\}. \tag{II.26}$$

We now look at the concavity of $g_{\mathrm{c}}(\mathbf{x})$. Since $h(\boldsymbol{\gamma}) \triangleq \log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T|$ is concave and non-decreasing w.r.t.$\boldsymbol{\gamma}\succeq\mathbf{0}$, we have

$$\log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T| = \min_{\mathbf{z}\succeq\mathbf{0}}\mathbf{z}^T\boldsymbol{\gamma} - h^*(\mathbf{z}) \tag{II.27}$$

where $h^*(\mathbf{z})$ is the concave conjugate of $h(\boldsymbol{\gamma})$ and can be expressed as $h^*(\mathbf{z}) = \min_{\boldsymbol{\gamma}\succeq\mathbf{0}}\mathbf{z}^T\boldsymbol{\gamma} - \log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T|$. Thus using (II.27) we can express (II.26) as

$$\begin{aligned} g_{\mathrm{c}}(\mathbf{x}) &= \min_{\boldsymbol{\gamma},\mathbf{z}\succeq\mathbf{0}}\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \mathbf{z}^T\boldsymbol{\gamma} - h^*(\mathbf{z}) \\ &= \min_{\boldsymbol{\gamma},\mathbf{z}\succeq\mathbf{0}}\sum_i\Big(\frac{\mathbf{x}_i^T\mathbf{B}_i^{-1}\mathbf{x}_i}{\gamma_i} + z_i\gamma_i\Big) - h^*(\mathbf{z}) \end{aligned} \tag{II.28}$$

Minimizing (II.28) over γ_i, we have

$$\gamma_i = z_i^{-\frac{1}{2}} \sqrt{\mathbf{x}_i^T \mathbf{B}_i^{-1} \mathbf{x}_i}, \quad \forall i \tag{II.29}$$

Plugging it in (II.28) leads to

$$g_{\mathrm{c}}(\mathbf{x}) = \min_{\mathbf{z} \succeq \mathbf{0}} \sum_i \left(2 z_i^{\frac{1}{2}} \sqrt{\mathbf{x}_i^T \mathbf{B}_i^{-1} \mathbf{x}_i} \right) - h^*(\mathbf{z}). \tag{II.30}$$

Using (II.30), the problem (II.25) now becomes:

$$\mathbf{x} \;=\; \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \Big[\min_{\mathbf{z} \succeq \mathbf{0}} \sum_i \left(2 z_i^{\frac{1}{2}} \sqrt{\mathbf{x}_i^T \mathbf{B}_i^{-1} \mathbf{x}_i} \right) - h^*(\mathbf{z}) \Big]. \tag{II.31}$$

To further simplify the expression, we now calculate the optimal values of $z_i^{\frac{1}{2}}$. However, we do not need to calculate the optimal values from the above expression. According to the duality property, from the relation (II.27) we can directly obtain the optimal value of $z_i^{1/2}$ as follows:

$$\begin{aligned} z_i^{\frac{1}{2}} &= \Big(\frac{\partial \log |\lambda \mathbf{I} + \mathbf{\Phi} \mathbf{\Sigma}_0 \mathbf{\Phi}^T|}{\partial \gamma_i} \Big)^{\frac{1}{2}} \\ &= \Big(\mathrm{Tr} \big[\mathbf{B}_i \mathbf{\Phi}^{i^T} \big(\lambda \mathbf{I} + \mathbf{\Phi} \mathbf{\Sigma}_0 \mathbf{\Phi}^T \big)^{-1} \mathbf{\Phi}^i \big] \Big)^{\frac{1}{2}}. \end{aligned} \tag{II.32}$$

Note that z_i is a function of $\boldsymbol{\gamma}$, while according to (II.29) γ_i is a function of $\mathbf{x}_i$ (and z_i). This means that the problem (II.31) should be solved in an iterative way. In the k-th iteration, once having used the learning rules (II.29) and (II.32) to obtain $(z_i^{(k)})^{1/2}$, we need to solve the following optimization problem:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \; \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \sum_i w_i^{(k)} \sqrt{\mathbf{x}_i^T \mathbf{B}_i^{-1} \mathbf{x}_i}, \tag{II.33}$$

where $w_i^{(k)} \triangleq 2 (z_i^{(k)})^{1/2}$. And the resulting $\mathbf{x}^{(k+1)}$ will be used to update γ_i and z_i for calculating the solution in the next iteration.

The solution to (II.33) can be calculated using any group-Lasso type algorithms. To see this, letting $\mathbf{u}_i \triangleq w_i^{(k)} \mathbf{B}_i^{-1/2} \mathbf{x}_i$, $\mathbf{u} \triangleq [\mathbf{u}_1^T, \cdots, \mathbf{u}_g^T]^T$, and $\mathbf{H} \triangleq \mathbf{\Phi} \cdot \mathrm{diag}\{\mathbf{B}_1^{1/2}/w_1^{(k)}, \cdots, \mathbf{B}_g^{1/2}/w_g^{(k)}\}$, the problem (II.33) can be transformed to the following one

$$\mathbf{u}^{(k+1)} = \arg\min_{\mathbf{u}} \|\mathbf{y} - \mathbf{H}\mathbf{u}\|_2^2 + \lambda \sum_i \|\mathbf{u}_i\|_2. \tag{II.34}$$

Clearly, each iteration is a standard group-Lasso type problem, while the whole algorithm is an iterative reweighted algorithm.

In the above development we did not consider the learning rules for the regularizers λ and $\mathbf{B}_i$. In fact, their computation greatly benefits from this iterative reweighted form. Since each iteration is a group-Lasso type problem, the optimal value of λ can be automatically selected in the group Lasso framework [130]. Also, since each iteration provides a block-sparse solution, which is close to the true solution, $\mathbf{B}_i$ can be directly calculated from the solution of the previous iteration. In particular, each non-zero block in the previous solution can be treated as a segment of AR(1) process, and its AR coefficient is thus estimated. The AR coefficients associated with all the non-zero blocks are averaged [2], and the average value, denoted by $\bar{r}$, is used to construct each $\widehat{\mathbf{B}}_i$ according to (II.17).

The above algorithm is denoted by **BSBL-ℓ_1**. Now we discuss the connection of BSBL-ℓ_1 to existing algorithms. When we do not consider the intra-block correlation (i.e. setting $\mathbf{B}_i = \mathbf{I}(\forall i)$) and also do not iterate the algorithm, BSBL-ℓ_1 reduces to the canonical group Lasso algorithm [160]. When we iterate the algorithm but ignore the intra-block correlation, the algorithm reduces to the block version of the iterative reweighted ℓ_1 algorithm [17].

In fact, BSBL-ℓ_1 can be seen as a hybrid of BSBL algorithms and group-Lasso type algorithms. From one side, it has the ability to adaptively learn and exploit the intra-block correlation for better performance, as BSBL-EM and BSBL-BO. From the other side, since it only takes few iterations (generally about 2 to 5 iterations in noisy environments) and each iteration can be implemented by any efficient group-Lasso type algorithm, it is much faster and is especially suitable for large-scale datasets, compared to BSBL-EM and BSBL-BO.

The algorithm also provides insights if we want to equip group-Lasso type algorithms with the ability to exploit intra-block correlation for better recovery performance. We can consider this iterative reweighted method and change the ℓ_2

[2]The averaging is necessary. Otherwise, the algorithm may have poor performance.

norm of $\mathbf{x}_i$, i.e., $\|\mathbf{x}_i\|_2$, to the Mahalanobis-distance type measure $\sqrt{\mathbf{x}_i^T \mathbf{B}_i^{-1} \mathbf{x}_i}$.

II.B.4 Remarks on BSBL-EM, BSBL-BO, and BSBL-ℓ_1

Generally, BSBL-EM has better recovery performance than the other two. But its speed is slower due to the use of the EM method.

BSBL-ℓ_1 is much faster than BSBL-EM due to the use of Group-Lasso type algorithms. Its speed is determined by a used Group-Lasso type algorithm. Different Group-Lasso type algorithms and software packages [137, 44, 26, 80, 31, 25] can result in different speed of BSBL-ℓ_1. Also, it may inherit some other characteristics of the used Group-Lasso type algorithm. For example, when BSBL-ℓ_1 uses the SLEP software [80] or the SPGL1 software [137] to compute Group-Lasso solutions in noiseless situations, it cannot obtain good recovery performance. In contrast, using the CVX software [26] in noiseless situations can achieve satisfactory performance, even better than BSBL-EM and BSBL-BO (see Section II.D.1). Therefore, one needs to be careful when choosing a Group-Lasso type algorithm for BSBL-ℓ_1 in specific applications.

BSBL-BO has a good balance between speed and recovery performance. It is generally slower than BSBL-ℓ_1 but much faster than BSBL-EM, while its recovery performance is slightly poorer than BSBL-EM but better than BSBL-ℓ_1.

Among the three algorithms, BSBL-EM and BSBL-BO has the ability to directly recover non-sparse signals or signals with non-sparse representation, as long as the non-sparse signals or the representation coefficients have correlation structures. Thus, both BSBL-EM and BSBL-BO can be used for compressed sensing of raw physiological signals for energy-efficient wireless telemonitoring [167, 168, 169]. In contrast, BSBL-ℓ_1 does not have satisfactory performance in this application.

There is another algorithm [79] derived from the framework using a fast marginal likelihood method [133]. It has slightly poorer recovery performance than BSBL-EM and BSBL-BO, but has much faster speed.

II.C Algorithms for the Situation When the Block Partition is Unknown

Now we extend the previous framework to derive algorithms when block partition is unknown. For the algorithm development, we assume that all the blocks are of equal size h and that the non-zeros blocks are arbitrarily located. Later we will see that the approximation of equal block-size is not limiting. Note that though the resulting algorithms are not very sensitive to the choice of h, algorithmic performance can be further improved if a suitable value of h is selected. We will comment more on h later.

This model is consistent with communication channel modeling where an ideal sparse channel consisting of a few specular multi-path components has a discrete-time, bandlimited, baseband representation, which exhibits a block sparse structure with the block centers determined by the arbitrary arrival times of the multi-path components. Since the blocks are arbitrarily located they can overlap giving rise to larger unequal blocks.

Given the identical block size h, there are $p \triangleq N - h + 1$ possible blocks in $\mathbf{x}$, which overlap each other. The i-th block starts at the i-th element of $\mathbf{x}$ and ends at the $(i + h - 1)$-th element. All the nonzero elements of $\mathbf{x}$ lie in some of these blocks. Similar to Section II.B, for the i-th block, we assume it satisfies a multivariate Gaussian distribution with the mean given by $\mathbf{0}$ and the covariance matrix given by $\gamma_i \mathbf{B}_i$, where $\mathbf{B}_i \in \mathbb{R}^{h \times h}$. So we have the prior of $\mathbf{x}$ as the form: $p(\mathbf{x}) \sim \mathcal{N}_x(\mathbf{0}, \mathbf{\Sigma}_0)$. Note that due to the overlapping locations of these blocks, $\mathbf{\Sigma}_0$ is no longer a block diagonal matrix. It has the structure that each $\gamma_i \mathbf{B}_i$ lies along the principal diagonal of $\mathbf{\Sigma}_0$ and overlaps other $\gamma_j \mathbf{B}_j$ in neighbor. Thus, we cannot directly use the previous BSBL framework and need to make some modifications.

To facilitate the use of the BSBL framework, we expand the covariance matrix $\mathbf{\Sigma}_0$ as follows:

$$\widetilde{\mathbf{\Sigma}}_0 = \text{diag}\{\gamma_1 \mathbf{B}_1, \cdots, \gamma_p \mathbf{B}_p\} \in \mathbb{R}^{ph \times ph} \tag{II.35}$$

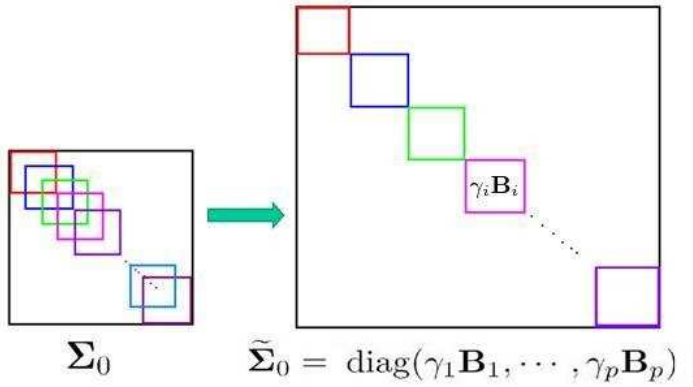


Figure II.1 Structures of the original $\boldsymbol{\Sigma}_0$ and the expanded $\widetilde{\boldsymbol{\Sigma}}_0$. Each color block corresponds to a possible nonzero block in $\mathbf{x}$.

Note that now $\gamma_i\mathbf{B}_i$ does not overlap other $\gamma_j\mathbf{B}_j (i \neq j)$. Figure II.1 shows the structures of the original $\boldsymbol{\Sigma}_0$ and the expanded $\widetilde{\boldsymbol{\Sigma}}_0$. The $\widetilde{\boldsymbol{\Sigma}}_0$ implies the decomposition of $\mathbf{x}$

$$\mathbf{x} = \sum_{i=1}^{p} \mathbf{E}_i \mathbf{z}_i, \tag{II.36}$$

where $E\{\mathbf{z}_i\} = \mathbf{0}$, $E\{\mathbf{z}_i\mathbf{z}_j^T\} = \delta_{i,j}\gamma_i\mathbf{B}_i$ ($\delta_{i,j} = 1$ if $i = j$; otherwise, $\delta_{i,j} = 0$), and $\mathbf{z} \triangleq [\mathbf{z}_1^T, \cdots, \mathbf{z}_p^T]^T \sim \mathcal{N}_z(\mathbf{0}, \widetilde{\boldsymbol{\Sigma}}_0)$. $\mathbf{E}_i \in \mathbb{R}^{N\times h}$ is a zero matrix except that the part from its i-th row to $(i + h - 1)$-th row is replaced by the identity matrix $\mathbf{I}$. Then the original model (II.1) can be expressed as:

$$\mathbf{y} = \sum_{i=1}^{p} \boldsymbol{\Phi}\mathbf{E}_i\mathbf{z}_i + \mathbf{v} \triangleq \mathbf{A}\mathbf{z} + \mathbf{v}, \tag{II.37}$$

where $\mathbf{A} \triangleq [\mathbf{A}_1, \cdots, \mathbf{A}_p]$ with $\mathbf{A}_i \triangleq \boldsymbol{\Phi}\mathbf{E}_i$. Now we see the new model (II.37) is exactly a BSBL model.

Therefore, following the development of BSBL-EM, BSBL-BO, and BSBL-ℓ_1, we can derive algorithms for this expanded model, which are called **EBSBL-EM**, **EBSBL-BO**, and **EBSBL-ℓ_1**, respectively.

In the above development we assumed that all the blocks have the same size h, which is known. However, this assumption is not crucial for practical use. When the size of a non-zero block of $\mathbf{x}$, say $\mathbf{x}_j$, is larger or equal to h, it can be recovered by a set of (overlapped) $\mathbf{z}_i$ ($i \in \mathcal{S}$, $\mathcal{S}$ is a non-empty set). When the size of $\mathbf{x}_j$ is smaller than h, it can be recovered by a $\mathbf{z}_i$ for some i. In this case, since $\mathbf{z}_i$ is larger, the elements in $\mathbf{z}_i$ with global locations (i.e., the indexes in $\mathbf{x}$) different from those of elements in $\mathbf{x}_j$ are very close to zero. In Section II.D.5, we will see different values of h lead to similar performance.

The above insight also implies that even if the block partition is unknown, one can partition a signal into a number of non-overlapping blocks with user-defined block sizes, and then perform the BSBL algorithms. But generally the BSBL algorithms are more sensitive to the block sizes than the EBSBL algorithms when recovering block sparse signals (see Section II.D.6) [3].

Note that our approach using the expanded model in the situation when block partition is unknown is quite different from existing approaches [159, 65, 101]. An advantage of our approach is that it simplifies the algorithm, which, in turn, increases robustness in noisy environments, as shown in Section II.D. Another benefit of this approach is that it facilitates the exploitation of intra-block correlation. Since the intra-block correlation widely exists in practical signals and exploiting such correlation can significantly improve performance, our approach is more competitive than existing approaches.

II.D Simulations

This section presents some representative experimental results based on computer simulations. Experimental results on real-world data can be found in later chapters.

Every set of experiment settings consisted of 400 trials. The matrix $\mathbf{\Phi}$

[3] When directly recovering non-sparse signals, the BSBL algorithms are not sensitive to the block sizes [167].

in all the experiments was generated as a zero mean random Gaussian matrix with columns normalized to unit ℓ_2 norm. In noisy experiments the Normalized Mean Square Error (NMSE) was used as a performance index, defined by $\|\widehat{\mathbf{x}} - \mathbf{x}_{\rm gen}\|_2^2/\|\mathbf{x}_{\rm gen}\|_2^2$, where $\widehat{\mathbf{x}}$ was the estimate of the true signal $\mathbf{x}_{\rm gen}$; in noiseless experiments the *success rate* was used as a performance index, defined as the percentage of successful trials in the 400 trials (A successful trial was defined as the one when NMSE $\leq 10^{-5}$).

In noiseless experiments, BSBL-ℓ_1 chose the Mixed ℓ_2/ℓ_1 Program (implemented by the CVX toolbox [26])to perform its every iteration; in noisy experiments, it chose the Group Basis Pursuit. For all of our algorithms, when calculating r, the formula $r \triangleq \text{sign}(\frac{m_1}{m_0})\min\{|\frac{m_1}{m_0}|, 0.99\}$, instead of the original formula $r = \frac{m_1}{m_0}$, was used to ensure the calculated r is feasible (not larger than 1 or smaller than -1). The same modification goes to $\bar{r}$.

The Matlab codes and demo files of BSBL algorithms and EBSBL algorithms can be downloaded at https://sites.google.com/site/researchbyzhang/bsbl , or http://dsp.ucsd.edu/~zhilin/BSBL.html .

II.D.1 Phase Transition

We first examined the empirical phase transitions [34] for our three BSBL algorithms, Block-OMP, Model-CoSaMP, the Mixed ℓ_2/ℓ_1 Program, and Group Basis Pursuit when exactly recovering a block sparse signal in noiseless environments. The phase transition is generally used to illustrate how sparsity level (defined as $\rho = K/M$, where K is the number of non-zero elements) and indeterminacy (defined as $\delta = M/N$) affect algorithms' success in exact recovery of sparse signals. Each point on the plotted phase transition curve corresponds to the success rate of an algorithm larger than or equal to 0.99 in 400 trials; above the curve the algorithm's success rate sharply drops, while below the curve the success rate is 1.

In the experiment the indeterminacy $\delta = M/N$ ran from 0.05 to 0.5 with N fixed to 1000. For each M and N, a block sparse signal was generated, which

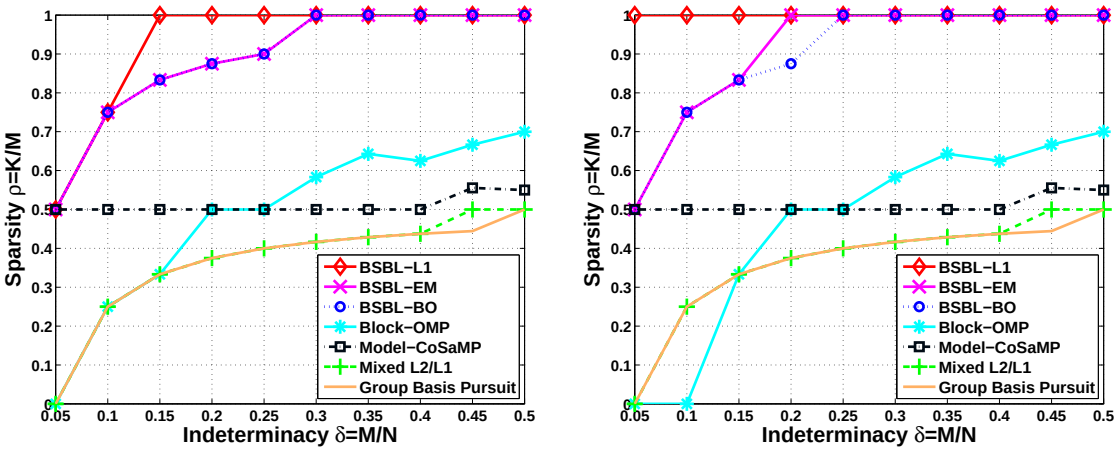
consisted of 40 blocks with identical block size 25. The number of non-zero blocks varied from 1 to 20, and thus the number of non-zero elements varied from 25 to 500. Locations of the non-zero blocks were determined randomly. The block partition was known to the algorithms, but the number of non-zero blocks and their locations were unknown to the algorithms. Each non-zero block was generated by a multivariate Gaussian distribution with zero mean and covariance matrix $\mathbf{\Sigma}_{\text{gen}}$. By changing the covariance matrix, thus changing intra-block correlation, we could study the effect of intra-block correlation on phase transitions of the algorithms.

We first considered the situation when there was no correlation within each non-zero block (i.e., $\mathbf{\Sigma}_{\text{gen}} = \mathbf{I}$). The empirical phase transition curves of all the algorithms are shown in Figure II.2 (a). Clearly, our three BSBL algorithms have the best performance. It is worth noting that when $\delta \geq 0.15$, BSBL-ℓ_1 exactly recovers block sparse signals with $\rho = 1$ with a high success rate (≥ 0.99).

The results become more interesting when intra-block correlation was 0.95 (i.e., $\mathbf{\Sigma}_{\text{gen}} = \text{Toeplitz}([1, 0.95, \cdots, 0.95^{24}])$). The empirical phase transition curves are shown in Figure II.2 (b), where all the three BSBL algorithms have improved performance. BSBL-ℓ_1 can exactly recover sparse signals with $\rho = 1$ even $\delta < 0.15$. And BSBL-EM and BSBL-BO now can exactly recover sparse signals with $\rho = 1$ when $\delta \geq 0.25$. Opposite to BSBL algorithms, the compared four algorithms show little change in performance when the intra-block correlation changes from 0 to 0.95.

These results are very interesting and surprising, since this may be the first time that an algorithm shows the ability to recover a block sparse signal of M non-zero elements from M measurements with a high success rate (≥ 0.99). Obviously, exploiting block structure and intra-block correlation plays a crucial role here. Further, these results indicate the advantages of the BSBL framework.

Figure II.3 (a) and (b) shows the empirical phase transitions of all the algorithms when elements of each non-zero block satisfy Bimodal Rayleigh distribution (i.e., $|\mathbf{x}_i|$ satisfies a Rayleigh distribution with the parameter $\sigma = 3$) and Laplacian



(a) Intra-Block Correlation: 0

(b) Intra-Block Correlation: 0.95

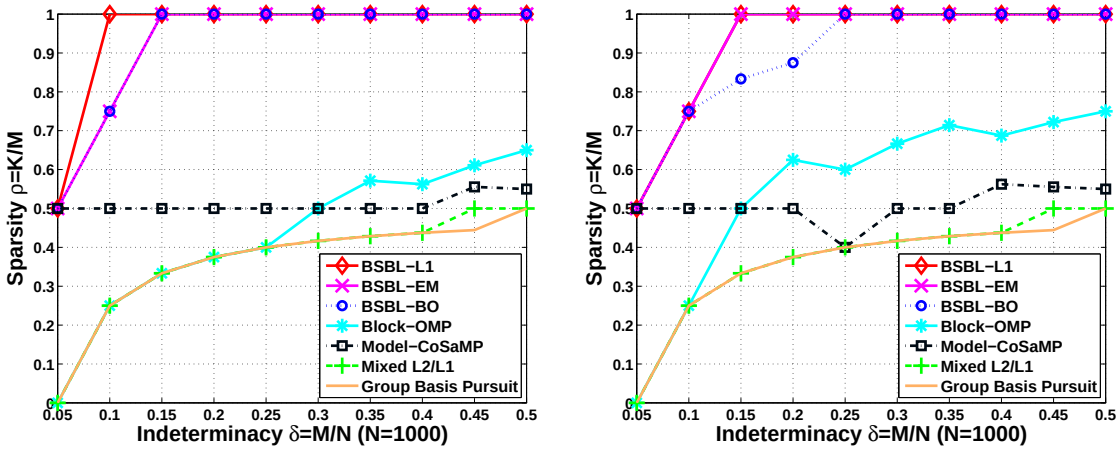
Figure II.2 Empirical 99% phase transitions of all the algorithms (a) when there was no correlation within each non-zero block, and (b) when the intra-block correlation was 0.95. Each point on a phase transition curve corresponds to the success rate larger than or equal to 0.99.

distribution (with zero mean and the scale parameter $b = 10$), respectively. We can see that the superiority of the BSBL algorithms is obvious over a wide range of distributions.

II.D.2 Benefit from Exploiting Intra-Block Correlation

The above results have given some hints on the benefit of exploiting intra-block correlation. To see this clearer, another noiseless experiment was carried out. The matrix $\mathbf{\Phi}$ was of the size 100×300. The signal consisted of 75 blocks with identical size. Only 20 of the blocks were nonzero. All the nonzero blocks had the same intra-block correlation (generated as in the above subsection), whose value ranged from -0.99 to 0.99. BSBL-EM, BSBL-BO and BSBL-L1 were performed in two ways. First, they adaptively learned and exploited the intra-block correlation. In the second case, they ignored the correlation (i.e. setting $\mathbf{B}_i = \mathbf{I}(\forall i)$).

The results are shown in Figure II.4 (a). First, we see that exploiting the



(a) Bimodal Rayleigh

(b) Laplacian

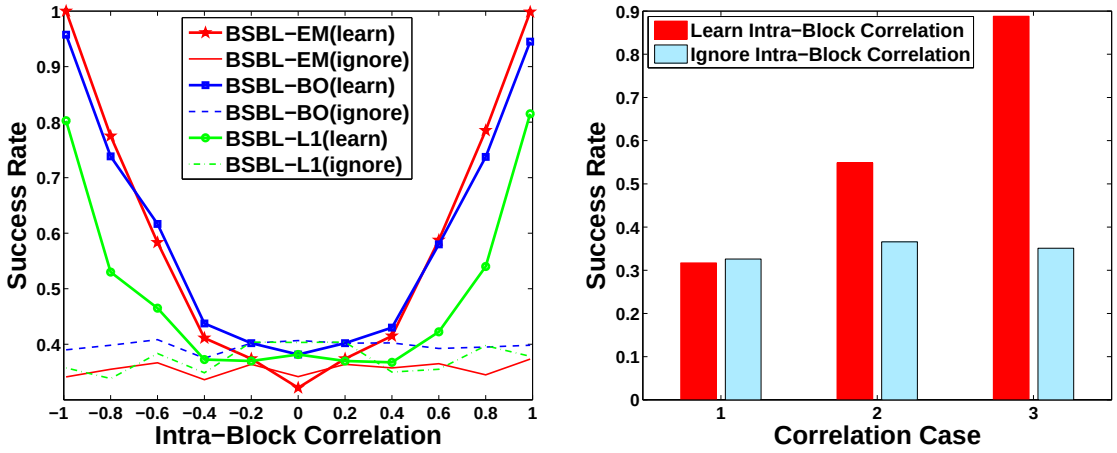
Figure II.3 Empirical 99% phase transitions of all the algorithms when elements in each non-zero block satisfied (a) a Bimodal Rayleigh distribution, and (b) a Laplacian distribution. Each point on a phase transition curve corresponds to the success rate larger than or equal to 0.99.

intra-block correlation greatly improved the performance of the BSBL algorithms. Second, when ignoring the intra-block correlation, the performance of the BSBL algorithms showed no obvious relation to the correlation [4]. In other words, no obvious negative effect is observed if ignoring the intra-block correlation. Note that the second observation is quite different from the observation on *temporal correlation* in MMV models [174], where we found that if temporal correlation is not exploited, algorithms have poorer performance with increasing temporal correlation values [5].

In the above experiment all the nonzero blocks had the same intra-block

[4]This phenomenon can also be observed from the performance of the compared algorithms in Section II.D.1, where their performance had little change when intra-block correlation dramatically varied.

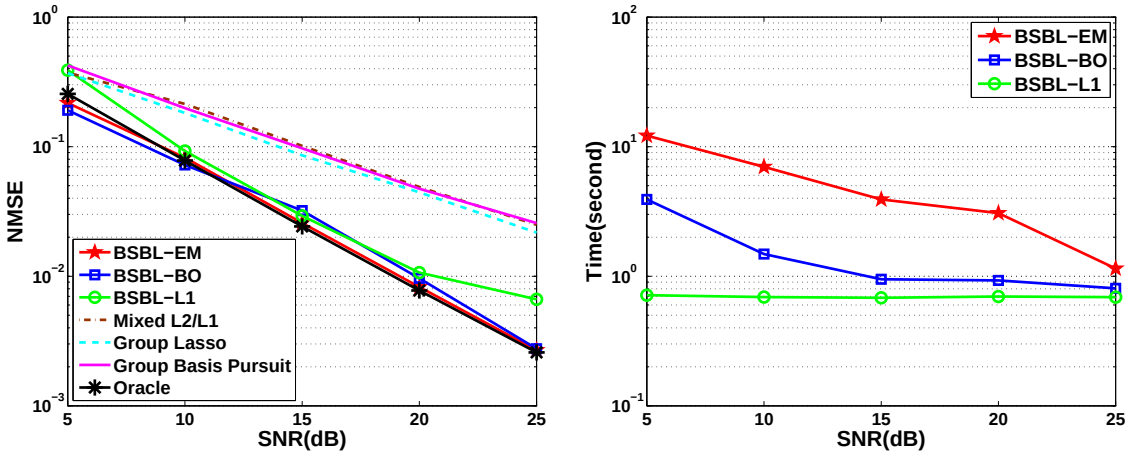
[5]The temporal correlation in an MMV model can be viewed as the intra-block correlation in the vectorized MMV model (which is a block sparse model). However, it should be noted that the sensing matrix in the vectorized MMV model has the specific structure $\mathbf{\Phi} \otimes \mathbf{I}_L$ [174], where $\mathbf{\Phi}$ is the sensing matrix in the original MMV model, $\otimes$ indicates the Kronecker product, $\mathbf{I}_L$ is the identity matrix with the dimension $L \times L$, and L is the number of measurement vectors in the MMV model. This structure is not present in the block sparsity model considered in this work and is believed to account for the different behavior with respect to the correlation.



(a) (b)

Figure II.4 (a) shows the benefit from exploiting intra-block correlation. (b) shows the performance of BSBL-EM in three correlation cases.

correlation. A natural question is: when different non-zero blocks have largely different intra-block correlation values, can our proposed algorithms still exploit the correlation to improve performance? To answer it, we considered three correlation cases. In the first case the intra-block correlation of each nonzero block was uniformly chosen from -1 to 1; in the second case, uniformly chosen from 0 to 1; and in the third case, uniformly chosen from 0.7 to 1. BSBL-EM was performed in two ways, i.e. exploiting correlation and ignoring correlation. The averaged results corresponding to the three cases are shown in Figure II.4 (b), indicated by 'Case 1', 'Case 2', and 'Case 3', respectively. We can see in Case 3 the benefit from exploiting the correlation is significant, while in Case 1 the benefit disappears (but exploiting correlation is not harmful). However, note that the Case 1 rarely happens in practice. In fact, in many practical problems the intra-block correlation of all nonzero blocks tends to be positive and high, which corresponds to Case 2 and Case 3.



(a) (b)

Figure II.5 (a) Performance comparison in different noise levels. (b) Speed comparison of the three BSBL algorithms in the noisy experiment.

II.D.3 Performance in Noisy Environments

We compared the three BSBL algorithms, Mixed ℓ_2/ℓ_1 Program, Group Lasso, and Group Basis Pursuit at different noise levels. In this experiment $M = 128$ and $N = 512$. The generated block sparse signal was partitioned into 64 blocks with identical block size 8. Seven blocks were non-zero, generated as in Section II.D.1. The intra-block correlation of each block varied from 0.8 to 1 randomly. Gaussian white noise was added so that the SNR, defined by $\text{SNR(dB)} \triangleq 20\log_{10}(\|\mathbf{\Phi x}_{\text{gen}}\|_2/\|\mathbf{v}\|_2)$, stepped from 5 dB to 25 dB for each generated signal. As a benchmark result, the 'oracle' result was calculated, which was the least-square estimate of $\mathbf{x}_{\text{gen}}$ given its true support.

The results are shown in Figure II.5 (a), from which we see our algorithms have much better performance, especially the performance curves of BSBL-EM and BSBL-BO almost overlap the 'Oracle' performance curve. Figure II.5 (b) gives the speed comparison of the three algorithms in a laptop with 2.8G Hz CPU and 6G RAM. Clearly, BSBL-ℓ_1 was the fastest among the three, due to the use of Group Basis Pursuit in its every iteration.

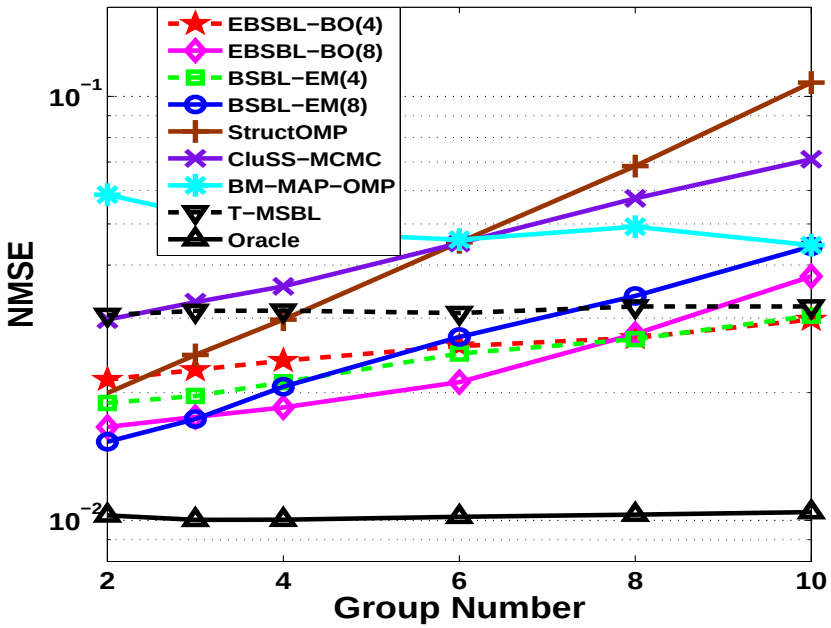


Figure II.6 Performance comparison in noisy environments (SNR=15 dB) when block partition was unknown.

II.D.4 Performance When Block Partition Is Unknown

We set up a noisy experiment when block partition was unknown and compared all of our algorithms with StructOMP (given the number of non-zero elements), BM-MAP-OMP (given the true noise variance), and CluSS-MCMC. The matrix $\mathbf{\Phi}$ was of the size 192×512. The signal $\mathbf{x}_{\text{gen}}$ had g_0 nonzero blocks with random size and random locations (not overlapping). g_0 was varied from 2 to 10. The total number of nonzero elements in $\mathbf{x}_{\text{gen}}$ was fixed to 48. The intra-block correlation in each block randomly varied from 0.8 to 1. SNR was 15 dB. As we stated in Section II.C, h is not crucial for practical use. To see this, we set $h = 4$ and $h = 8$ for our algorithms. But to prevent from being over-crowded when plotting performance curves, we only display BSBL-EM and EBSBL-BO with $h = 4$ and $h = 8$. We also performed T-MSBL [174] here. Note that when T-MSBL is used in the block sparse model, it can be viewed as a special case of BSBL-EM with each

block size being 1. The results are shown in Figure II.6. We see that our algorithms outperformed StructOMP, CluSS-MCMC, and BM-MAP-OMP, and that for either BSBL-EM or EBSBL-BO, setting $h = 4$ or $h = 8$ led to similar performance.

II.D.5 Effect of h on the Performance of EBSBL Algorithms

To better see the effect of h on the performance of EBSBL algorithms, another experiment was carried out, in which EBSBL-EM, T-MSBL, CluSS-MCMC and StructOMP were compared.

The matrix $\mathbf{\Phi}$ was of the size 80×256. The number of nonzero elements in the signal was fixed to 32, which were randomly put into 4 blocks. So each block had random size and random location. To more clearly see the effectiveness of our generalization model, we set intra-block correlation to zero. SNR was 25 dB. For EBSBL-EM, we set h to different values ranging from 2 to 10. The result (Figure II.7) shows that EBSBL-EM had much better performance than other compared algorithms. Its performance was only slightly changed when h chose values from 3 to 10.

II.D.6 Compare BSBL and EBSBL in the Situation When the Block Partition is Unknown

Before we pointed out that in the situation when the block partition is unknown, we can use both EBSBL algorithms and BSBL algorithms with a user-defined h. But one should be aware that when recovering a block sparse signal in this situation, BSBL algorithms are more sensitive to h than EBSBL algorithms, especially when the number of zero elements between two non-zero blocks is smaller than h.

To see this, we compared BSBL-EM and EBSBL-EM using the previous experiment setting. For both algorithms, we set h to different values ranging from 2 to 10. For each value of h, the experiment was repeated 100 trials. The averaged NMSE of both algorithms was shown in Figure II.8. We can see the performance

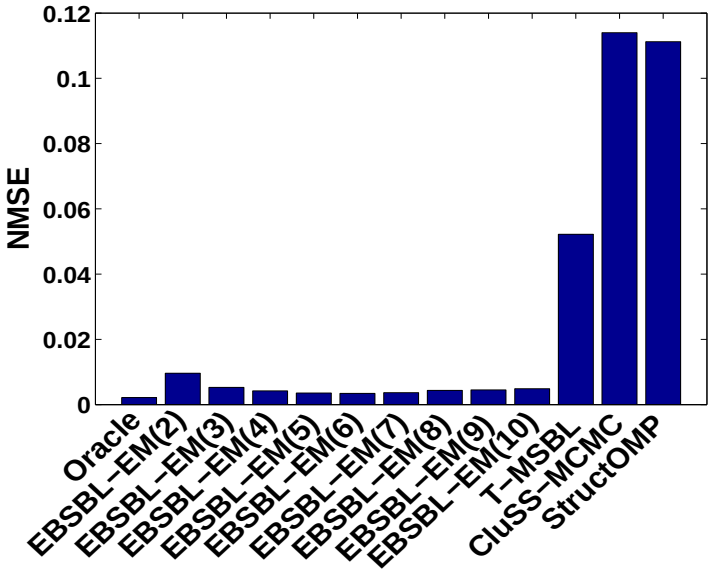


Figure II.7 Effect of h on the EBSBL-EM algorithm when h varied from 2 to 10. The label 'EBSBL-EM(k)' denotes EBSBL-EM with $h = k$.

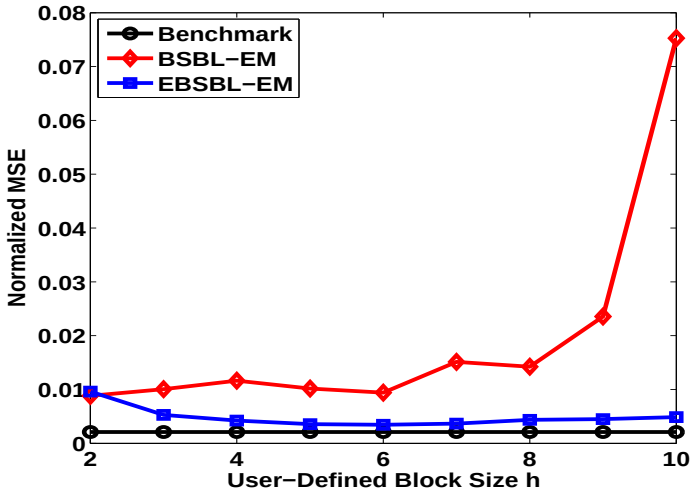


Figure II.8 Comparison between BSBL-EM and EBSBL-EM in the situation when the block partition is unknown.

of BSBL-EM changed dramatically when h changed.

II.E Conclusion

Based on the block sparse Bayesian learning framework and its extension, in this chapter we proposed a number of algorithms to recover block sparse signals when their block structure is known or is unknown. These algorithms have the ability to explore and exploit intra-block correlation in signals for better performance. Experiments showed that these algorithms significantly outperform existing algorithms. The derived algorithms also suggest that the iterative reweighted framework is a promising method for Group-Lasso type algorithms to exploit intra-block correlation.

II.F Acknowledgements

The text of Chapter II, in full, is a reprint of the material as it appears in: Zhilin Zhang and Bhaskar D. Rao, "Extension of SBL Algorithms for the Recovery of Block Sparse Signals with Intra-Block Correlation", to appear in IEEE Trans. on Signal Processing. The dissertation author was a primary researcher and author of the cited material.

Chapter III

Sparse Bayesian Learning Exploiting Temporal Correlation

Motivated by many applications such as EEG/MEG source localization and direction-of-arrival (DOA) estimation, where a sequence of measurement vectors are available, the basic SMV model (II.1) has been extended to the multiple measurement vector (MMV) model in [106, 25], given by

$$\mathbf{Y} = \mathbf{\Phi X} + \mathbf{V}, \tag{III.1}$$

where $\mathbf{Y} \triangleq [\mathbf{Y}_{\cdot 1}, \cdots, \mathbf{Y}_{\cdot L}] \in \mathbb{R}^{M \times L}$ is an available measurement matrix consisting of L measurement vectors, $\mathbf{X} \triangleq [\mathbf{X}_{\cdot 1}, \cdots, \mathbf{X}_{\cdot L}] \in \mathbb{R}^{N \times L}$ is an unknown source matrix (or called a solution matrix) with each row representing a possible source [1], and $\mathbf{V}$ is an unknown noise matrix. A key assumption in the MMV model is that the support (i.e. indexes of nonzero entries) of every column in $\mathbf{X}$ is identical (referred as *the common sparsity assumption* [25]). In addition, similar to the constraint in the SMV model, the number of nonzero rows in $\mathbf{X}$ has to be below a threshold to ensure a unique and global solution [25]. This leads to the fact that $\mathbf{X}$ has a small number of nonzero rows.

It has been shown that compared to the SMV case, the successful recovery rate can be greatly improved using multiple measurement vectors [25, 44, 45, 73]. For example, Cotter and Rao [25] showed that by taking advantage of the MMV formulation, one can relax the upper bound in the uniqueness condition for the solution. Tang, Eldar and their colleagues [128, 45] showed that under certain mild assumptions the recovery rate increases exponentially with the number of measurement vectors L. Jin and Rao [73, 74] analyzed the benefits of increasing L by relating the MMV model to the capacity regions of MIMO communication channels. All these theoretical results reveal the advantages of the MMV model and support increasing L for better recovery performance.

However, under the common sparsity assumption we cannot obtain many measurement vectors in practical applications. The main reason is that the sparsity profile of practical signals is (slowly) time-varying, so the common sparsity assump-

[1] Here for convenience we call each row in $\mathbf{X}$ a source. The term is often used in application-oriented literature. The i-th source is denoted by $\mathbf{X}_{i\cdot}$.

tion is valid for only a small L in the MMV model. For example, in EEG/MEG source localization there is considerable evidence [92] that a given pattern of dipole-source distributions [2] may only exist for 10-20 ms. Since the EEG/MEG sampling frequency is generally 250 Hz, a dipole-source pattern may only exist through 5 snapshots (i.e. in the MMV model $L = 5$). In DOA estimation [24], directions of targets [3] are continuously changing, and thus the source vectors that satisfy the common sparsity assumption are few. Of course, one can increase the measurement vector number at the cost of increasing the source number, but a larger source number can result in degraded recovery performance.

Thanks to numerous algorithms for the basic SMV model, most MMV algorithms are obtained by straightforward extension of the SMV algorithms; for example, calculating the ℓ_2 norm of each row of $\mathbf{X}$, forming a vector, and then imposing the sparsity constraint on the vector. These algorithms can be roughly divided into greedy algorithms [136, 77], algorithms based on mixed norm optimization [100, 135, 8, 66, 115], iterative reweighted algorithms [149, 25], messaging passing algorithms [178, 116], and Bayesian algorithms [174, 173, 172, 154].

But for tractability purposes, most existing MMV algorithms (and theoretical works) assume that sources are independent and identically distributed (i.i.d.) processes. This contradicts real-world scenarios, since a practical source often has strong temporal correlation. For example, waveform smoothness of biomedical signals has been exploited in signal processing for several decades. Besides, due to high sampling frequency, amplitudes of successive samplings of a source are strongly correlated. Recently, Zdunek and Cichocki [163] proposed the SOB-MFOCUSS algorithm, which exploits the waveform smoothness via a pre-defined smoothness matrix. However, the design of the smoothness matrix is completely subjective and not data-adaptive. In fact, in the task of sparse signal recovery, learning temporal correlation of a source is a difficult problem. Generally, such

[2] In this application the set of indexes of nonzero rows in $\mathbf{X}$ is called a pattern of dipole-source distribution.

[3] In this application the index of a nonzero row in $\mathbf{X}$ indicates a direction.

structures are learned via a training dataset (which often contains sufficient data without noise for robust statistical inference) [22, 68]. Although effective for some specific signals, this method is limited. Having noticed that temporal correlation strongly affects performance of existing algorithms, in [172] we derived the AR-SBL algorithm, which models each source as a first-order autoregressive (AR) process and learns AR coefficients from data per se. Although the algorithm has superior performance compared to MMV algorithms in the presence of temporal correlation, it is slow, which limits its applications. As such, there is a need for efficient algorithms that can deal more effectively with temporal correlation.

Noticing the relation between the MMV model and the block sparse model, we first transform the MMV model into a block sparse model, where temporal correlation of sources can be easily modeled. Then in the block sparse model, we derive an SBL algorithm, called T-SBL, which is very effective but is slow due to its operation in a higher dimensional parameter space resulting from the MMV-to-SMV transformation. Thus, we make some approximations and derive two fast versions, called T-MSBL and T-MSBL-FP, respectively. T-MSBL is derived using the EM method, while T-MSBL-FP is derived using a fixed-point method. Both algorithms operate in the original parameter space. Similar to T-SBL, T-MSBL and T-MSBL-FP are also effective but has much lower computational complexity. Interestingly, when compared to the MSBL algorithm [154], the only change of T-MSBL is the replacement of $\|\mathbf{X}_{i\cdot}\|_2^2$ with the Mahalanobis distance measure, i.e. $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$, where $\mathbf{B}$ is a positive definite matrix estimated from data and can be partially interpreted as a covariance matrix. We analyze the global minimum and the local minima of the algorithms' cost function. One of the key results is that in the noiseless case the global minimum corresponds to the sparsest solution. Extensive experiments not only show the superiority of the proposed algorithms, but also provide some interesting (even counter-intuitive) phenomena that may motivate future theoretical study.

We introduce the notations used in this chapter:

- $\|\mathbf{x}\|_1, \|\mathbf{x}\|_2, \|\mathbf{A}\|_{\mathcal{F}}$ denote the ℓ_1 norm of the vector $\mathbf{x}$, the ℓ_2 norm of $\mathbf{x}$, and the Frobenius norm of the matrix $\mathbf{A}$, respectively. $\|\mathbf{A}\|_0$ and $\|\mathbf{x}\|_0$ denote the number of nonzero rows in the matrix $\mathbf{A}$ and the number of nonzero elements in the vector $\mathbf{x}$, respectively;

- Bold symbols are reserved for vectors and matrices. Particularly, $\mathbf{I}_L$ denotes the identity matrix with size $L \times L$. When the dimension is evident from the context, for simplicity, we just use $\mathbf{I}$;

- $\text{diag}\{a_1, \cdots, a_M\}$ denotes a diagonal matrix with principal diagonal elements being $a_1, \cdots, a_M$ in turn; if $\mathbf{A}_1, \cdots, \mathbf{A}_M$ are square matrices, then $\text{diag}\{\mathbf{A}_1, \cdots, \mathbf{A}_M\}$ denotes a block diagonal matrix with principal diagonal blocks being $\mathbf{A}_1, \cdots, \mathbf{A}_M$ in turn;

- For a matrix $\mathbf{A}$, $\mathbf{A}_{i\cdot}$ denotes the i-th row, $\mathbf{A}_{\cdot i}$ denotes the i-th column, and $\mathbf{A}_{i,j}$ denotes the element that lies in the i-th row and the j-th column;

- $\mathbf{A} \otimes \mathbf{B}$ represents the Kronecker product of the two matrices $\mathbf{A}$ and $\mathbf{B}$. $\text{vec}(\mathbf{A})$ denotes the vectorization of the matrix $\mathbf{A}$ formed by stacking its columns into a single column vector. $\text{Tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$. $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$.

III.A Problem Statement

Most existing works do not deal with temporal correlation of sources. For many non-Bayesian algorithms, incorporating temporal correlation is not easy due to the lack of a well defined methodology to modify the diversity measures employed in the optimization procedure. For example, it is not clear how to best incorporate correlation in ℓ_1 norm based methods. For this reason, we adopt a probabilistic approach to incorporate correlation structure. Particularly, we have found it convenient to incorporate correlation into the SBL methodology.

To exploit temporal correlation, we use the block sparse Bayesian learning framework stated in the previous chapter. To use this framework, the MMV model is transformed to a block SMV model. In this way, we can easily model the temporal correlation of sources and derive new algorithms.

First, we assume all the sources $\mathbf{X}_{i\cdot}$ $(\forall i)$ are mutually independent, and the density of each $\mathbf{X}_{i\cdot}$ is (parameterized) Gaussian, given by

$$p(\mathbf{X}_{i\cdot};\gamma_i,\mathbf{B}_i)\sim\mathcal{N}(\mathbf{0},\gamma_i\mathbf{B}_i),\quad i=1,\cdots,M$$

where γ_i is a nonnegative hyperparameter controlling the row sparsity of $\mathbf{X}$. When $\gamma_i=0$, the associated $\mathbf{X}_{i\cdot}$ becomes zeros. $\mathbf{B}_i$ is a positive definite matrix that captures the correlation structure of $\mathbf{X}_{i\cdot}$ and needs to be estimated.

By letting $\mathbf{y}=\mathrm{vec}(\mathbf{Y}^T)\in\mathbb{R}^{ML\times 1}$, $\mathbf{D}=\mathbf{\Phi}\otimes\mathbf{I}_L$, $\mathbf{x}=\mathrm{vec}(\mathbf{X}^T)\in\mathbb{R}^{NL\times 1}$, $\mathbf{v}=\mathrm{vec}(\mathbf{V}^T)$, we can transform the MMV model to the following block sparse model

$$\mathbf{y}=\mathbf{D}\mathbf{x}+\mathbf{v}. \tag{III.2}$$

Obviously, $\mathbf{x}$ is block-sparse.

Assume elements in the noise vector $\mathbf{v}$ are independent and each has a Gaussian distribution, i.e. $p(v_i)\sim\mathcal{N}(0,\lambda)$, where v_i is the i-th element in $\mathbf{v}$ and λ is the variance. For the block model (III.2), the Gaussian likelihood is

$$p(\mathbf{y}|\mathbf{x};\lambda)\sim\mathcal{N}_{y|x}(\mathbf{D}\mathbf{x},\lambda\mathbf{I}). \tag{III.3}$$

The prior for $\mathbf{x}$ is given by

$$p(\mathbf{x};\gamma_i,\mathbf{B}_i,\forall i)\sim\mathcal{N}_x(\mathbf{0},\mathbf{\Sigma}_0), \tag{III.4}$$

where $\mathbf{\Sigma}_0$ is

$$\mathbf{\Sigma}_0=\begin{bmatrix}\gamma_1\mathbf{B}_1 & & \\ & \ddots & \\ & & \gamma_N\mathbf{B}_N\end{bmatrix}. \tag{III.5}$$

Using Bayes' rule we obtain the posterior density of $\mathbf{x}$, which is also Gaussian,

$$p(\mathbf{x}|\mathbf{y};\lambda,\gamma_i,\mathbf{B}_i,\forall i) = \mathcal{N}_x(\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x) \tag{III.6}$$

with the mean

$$\boldsymbol{\mu}_x = \frac{1}{\lambda}\boldsymbol{\Sigma}_x\mathbf{D}^T\mathbf{y} \tag{III.7}$$

and the covariance matrix

$$\begin{aligned}\boldsymbol{\Sigma}_x &= (\boldsymbol{\Sigma}_0^{-1} + \frac{1}{\lambda}\mathbf{D}^T\mathbf{D})^{-1} &\text{(III.8)}\\ &= \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\mathbf{D}^T\big(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\big)^{-1}\mathbf{D}\boldsymbol{\Sigma}_0. &\text{(III.9)}\end{aligned}$$

So given all the parameters $\lambda,\gamma_i,\mathbf{B}_i,\forall i$, the MAP estimate of $\mathbf{x}$ is given by:

$$\begin{aligned}\mathbf{x}^* \triangleq \boldsymbol{\mu}_x &= (\lambda\boldsymbol{\Sigma}_0^{-1} + \mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{y}\\ &= \boldsymbol{\Sigma}_0\mathbf{D}^T\big(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\big)^{-1}\mathbf{y} &\text{(III.10)}\end{aligned}$$

where the last equation follows the matrix identity $(\mathbf{I}+\mathbf{AB})^{-1}\mathbf{A} \equiv \mathbf{A}(\mathbf{I}+\mathbf{BA})^{-1}$, and $\boldsymbol{\Sigma}_0$ is the block diagonal matrix given by (III.5) with many diagonal block matrices being zeros. Clearly, the block sparsity of $\mathbf{x}^*$ is controlled by the γ_i's in $\boldsymbol{\Sigma}_0$: during the learning procedure, when $\gamma_k = 0$, the associated k-th block in $\mathbf{x}^*$ becomes zeros [4].

To estimate the parameters we can use evidence maximization or Type-II maximum likelihood [132]. This involves marginalizing over the weights $\mathbf{x}$ and then performing maximum likelihood estimation. Note that the whole framework including the solution (III.10) and the parameter estimation is the BSBL framework. Note that in contrast to the original SBL framework, the BSBL framework models the temporal structures of sources in the prior density via the matrices $\mathbf{B}_i$ $(i = 1,\cdots,N)$. Different ways to learn the matrices result in different algorithms. We will discuss the learning of these matrices and other parameters in the following sections.

[4] In practice, we judge whether γ_k is less than a small threshold, e.g. 10^{-3}. If it is, then the associated dictionary vectors are pruned out from the learning procedure and the associated block in $\mathbf{x}$ is set to zeros.

III.B Algorithm Development

III.B.1 T-SBL: SBL Exploiting Temporal Correlation

Before estimating the parameters, we note that assigning a different matrix $\mathbf{B}_i$ to each source $\mathbf{X}_{i\cdot}$ will result in overfitting. To avoid the overfitting, we consider using one positive definite matrix $\mathbf{B}$ to model all the source covariance matrices up to a scalar [5]. Thus Eq.(III.5) becomes $\mathbf{\Sigma}_0 = \mathbf{\Gamma} \otimes \mathbf{B}$ with $\mathbf{\Gamma} \triangleq \text{diag}(\gamma_1, \cdots, \gamma_N)$. Although this strategy is equivalent to assuming all the sources have the same correlation structure, it leads to very good results even if all the sources have totally different correlation structures (see the simulations in Section III.E). More importantly, this constraint does not destroy the global minimum property (i.e. the global unique solution is the sparsest solution) of our algorithms, as shown in Section III.D.

To find the parameters $\Theta = \{\gamma_1, \cdots, \gamma_M, \mathbf{B}, \lambda\}$, we employ the EM method to maximize $p(\mathbf{y}; \Theta)$. This is equivalent to minimizing $-\log p(\mathbf{y}; \Theta)$, yielding the effective cost function:

$$\begin{aligned} \mathcal{L}(\Theta) &\triangleq -2 \log \int p(\mathbf{y}|\mathbf{x}; \lambda) p(\mathbf{x}; \gamma_i, \mathbf{B}_i, \forall i) d\mathbf{x} \\ &= \mathbf{y}^T (\mathbf{\Sigma}_y)^{-1} \mathbf{y} + \log |\mathbf{\Sigma}_y|, \end{aligned} \tag{III.11}$$

where $\mathbf{\Sigma}_y \triangleq \lambda \mathbf{I} + \mathbf{D} \mathbf{\Sigma}_0 \mathbf{D}^T$. The EM formulation proceeds by treating $\mathbf{x}$ as hidden variables and then maximizing:

$$\begin{aligned} Q(\Theta) &= E_{x|y;\Theta^{(\text{old})}} \big[\log p(\mathbf{y}, \mathbf{x}; \Theta) \big] \\ &= E_{x|y;\Theta^{(\text{old})}} \big[\log p(\mathbf{y}|\mathbf{x}; \lambda) \big] \\ &\quad + E_{x|y;\Theta^{(\text{old})}} \big[\log p(\mathbf{x}; \gamma_1, \cdots, \gamma_N, \mathbf{B}) \big] \end{aligned} \tag{III.12}$$

where $\Theta^{(\text{old})}$ denotes the estimated hyperparameters in the previous iteration.

To estimate $\boldsymbol{\gamma} \triangleq [\gamma_1, \cdots, \gamma_N]$ and $\mathbf{B}$, we notice that the first term in (III.12) is unrelated to $\boldsymbol{\gamma}$ and $\mathbf{B}$. So, the Q function (III.12) can be simplified to:

$$Q(\boldsymbol{\gamma}, \mathbf{B}) = E_{x|y;\Theta^{(\text{old})}} \big[\log p(\mathbf{x}; \boldsymbol{\gamma}, \mathbf{B}) \big]. \tag{III.13}$$

[5] Note that the covariance matrix in the density of $\mathbf{X}_{i\cdot}$ is $\gamma_i \mathbf{B}_i$.

It can be shown that

$$\log p(\mathbf{x};\boldsymbol{\gamma},\mathbf{B}) \propto -\frac{1}{2}\log\big(|\boldsymbol{\Gamma}|^L|\mathbf{B}|^N\big) - \frac{1}{2}\mathbf{x}^T(\boldsymbol{\Gamma}^{-1}\otimes\mathbf{B}^{-1})\mathbf{x}, \tag{III.14}$$

which results in

$$\begin{aligned} Q(\boldsymbol{\gamma},\mathbf{B}) \;\; \propto \;\; & -\frac{L}{2}\log\big(|\boldsymbol{\Gamma}|\big) - \frac{N}{2}\log\big(|\mathbf{B}|\big) \\ & -\frac{1}{2}\mathrm{Tr}\Big[\big(\boldsymbol{\Gamma}^{-1}\otimes\mathbf{B}^{-1}\big)\big(\boldsymbol{\Sigma}_x+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^T\big)\Big], \end{aligned} \tag{III.15}$$

where $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are evaluated according to (III.7) and (III.9), given the estimated parameters $\Theta^{(\mathrm{old})}$.

The derivative of (III.15) with respect to γ_i $(i=1,\cdots,N)$ is given by

$$\frac{\partial Q}{\partial\gamma_i} = -\frac{L}{2\gamma_i} + \frac{1}{2\gamma_i^2}\mathrm{Tr}\Big[\mathbf{B}^{-1}\big(\boldsymbol{\Sigma}_x^i+\boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T\big)\Big], \tag{III.16}$$

where we define (using the MATLAB notations)

$$\begin{cases} \boldsymbol{\mu}_x^i \triangleq \boldsymbol{\mu}_x((i-1)L+1\;:\;iL) \\ \boldsymbol{\Sigma}_x^i \triangleq \boldsymbol{\Sigma}_x((i-1)L+1\;:\;iL\;,\;(i-1)L+1\;:\;iL) \end{cases} \tag{III.17}$$

So the learning rule for γ_i $(i=1,\cdots,N)$ is given by

$$\gamma_i \leftarrow \frac{\mathrm{Tr}\big[\mathbf{B}^{-1}\big(\boldsymbol{\Sigma}_x^i+\boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T\big)\big]}{L}, \quad i=1,\cdots,M \tag{III.18}$$

On the other hand, the gradient of (III.15) over $\mathbf{B}$ is given by

$$\frac{\partial Q}{\partial\mathbf{B}} = -\frac{N}{2}\mathbf{B}^{-1} + \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\gamma_i}\mathbf{B}^{-1}\big(\boldsymbol{\Sigma}_x^i+\boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T\big)\mathbf{B}^{-1}. \tag{III.19}$$

Thus we obtain the learning rule for $\mathbf{B}$:

$$\mathbf{B} \leftarrow \frac{1}{N}\sum_{i=1}^{N}\frac{\boldsymbol{\Sigma}_x^i+\boldsymbol{\mu}_x^i(\boldsymbol{\mu}_x^i)^T}{\gamma_i}. \tag{III.20}$$

To estimate λ, the Q function (III.12) can be simplified to

$$\begin{aligned}
Q(\lambda) &= E_{x|y;\Theta^{(\text{old})}}\big[\log p(\mathbf{y}|\mathbf{x};\lambda)\big] \\
&\propto -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}E_{x|y;\Theta^{(\text{old})}}\big[\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2\big] \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\Big[\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2 + E_{x|y;\Theta^{(\text{old})}}\big[\|\mathbf{D}(\mathbf{x}-\boldsymbol{\mu}_x)\|_2^2\big]\Big] \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\Big[\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2 + \mathrm{Tr}\big(\boldsymbol{\Sigma}_x\mathbf{D}^T\mathbf{D}\big)\Big] \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\Big[\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2 + \widehat{\lambda}\mathrm{Tr}\big(\boldsymbol{\Sigma}_x(\boldsymbol{\Sigma}_x^{-1}-\boldsymbol{\Sigma}_0^{-1})\big)\Big] \quad \text{(III.21)} \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\Big[\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2 + \widehat{\lambda}\big[NL - \mathrm{Tr}(\boldsymbol{\Sigma}_x\boldsymbol{\Sigma}_0^{-1})\big]\Big], \quad \text{(III.22)}
\end{aligned}$$

where (III.21) follows from the first equation in (III.9), and $\widehat{\lambda}$ denotes the estimated λ in the previous iteration. The λ learning rule is obtained by setting the derivative of (III.22) over λ to zero, leading to

$$\lambda \leftarrow \frac{\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2 + \lambda\big[NL - \mathrm{Tr}(\boldsymbol{\Sigma}_x\boldsymbol{\Sigma}_0^{-1})\big]}{ML}, \tag{III.23}$$

where the λ on the right-hand side is the $\widehat{\lambda}$ in (III.22). There are some challenges to estimate λ in SMV models. This, however, is alleviated in MMV models when considering temporal correlation. We elaborate on this next.

In the SBL framework (either for the SMV model or for the MMV model), many learning rules for λ have been derived [132, 153, 154, 105]. However, in noisy environments some of the learning rules probably cannot provide an optimal λ, thus leading to degraded performance. For the basic SBL/MSBL algorithms, Wipf et al [154] pointed out that the reason is that λ and appropriate M nonzero parameters γ_i make an identical contribution to the covariance $\boldsymbol{\Sigma}_y = \lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T$ in the cost functions of SBL/MSBL. To explain this, they gave an example: let a dictionary matrix $\boldsymbol{\Phi}' = [\boldsymbol{\Phi}_0, \mathbf{I}]$, where $\boldsymbol{\Phi}' \in \mathbb{R}^{M\times N}$ and $\boldsymbol{\Phi}_0 \in \mathbb{R}^{M\times(N-M)}$. Then the λ as well as the M parameters $\{\gamma_{N-M+1}, \cdots, \gamma_N\}$ associated with the columns of

the identity matrix in Φ' are not identifiable, because

$$\begin{aligned}\mathbf{\Sigma}_y &= \lambda\mathbf{I} + \mathbf{\Phi}'\mathbf{\Gamma}\mathbf{\Phi}'^T \\ &= \lambda\mathbf{I} + [\mathbf{\Phi}_0, \mathbf{I}]\text{diag}\{\gamma_1, \cdots, \gamma_N\}[\mathbf{\Phi}_0, \mathbf{I}]^T \\ &= \lambda\mathbf{I} + \mathbf{\Phi}_0\text{diag}\{\gamma_1, \cdots, \gamma_{N-M}\}\mathbf{\Phi}_0^T \\ &\quad +\text{diag}\{\gamma_{N-M+1}, \cdots, \gamma_N\}\end{aligned}$$

indicating a nonzero value of λ and appropriate values of the M nonzero parameters, namely $\gamma_{N-M+1}, \cdots, \gamma_N$, can make an identical contribution to the covariance matrix $\mathbf{\Sigma}_y$. This problem can be worse when the noise covariance matrix is $\text{diag}(\lambda_1, \cdots, \lambda_M)$ with arbitrary nonzero λ_i, instead of $\lambda\mathbf{I}$.

However, our learning rule (III.23) does not have such ambiguity problem. To see this, we now examine the covariance matrix $\mathbf{\Sigma}_y$ in our cost function (III.11). Noting that $\mathbf{D} = \mathbf{\Phi}' \otimes \mathbf{I}$, we have

$$\begin{aligned}\mathbf{\Sigma}_y &= \lambda\mathbf{I} + \mathbf{D}\mathbf{\Sigma}_0\mathbf{D}^T \\ &= \lambda\mathbf{I} + (\mathbf{\Phi}' \otimes \mathbf{I})\big(\text{diag}\{\gamma_1, \cdots, \gamma_N\} \otimes \mathbf{B}\big)(\mathbf{\Phi}' \otimes \mathbf{I})^T \\ &= \lambda\mathbf{I} + [\mathbf{\Phi}_0 \otimes \mathbf{I}, \mathbf{I} \otimes \mathbf{I}]\big(\text{diag}\{\gamma_1, \cdots, \gamma_N\} \otimes \mathbf{B}\big) \\ &\quad \cdot[\mathbf{\Phi}_0 \otimes \mathbf{I}, \mathbf{I} \otimes \mathbf{I}]^T \\ &= \lambda\mathbf{I} + (\mathbf{\Phi}_0\text{diag}\{\gamma_1, \cdots, \gamma_{N-M}\}\mathbf{\Phi}_0^T) \otimes \mathbf{B} \\ &\quad +\text{diag}\{\gamma_{N-M+1}, \cdots, \gamma_N\} \otimes \mathbf{B}.\end{aligned}$$

Obviously, since $\mathbf{B}$ is not an identity matrix [6], λ and $\{\gamma_{N-M+1}, \cdots, \gamma_N\}$ cannot identically contribute to $\mathbf{\Sigma}_y$.

The SBL algorithm using the learning rules (III.9), (III.10), (III.18), (III.20) and (III.23) is denoted by **T-SBL**.

[6] Note that even all the sources are i.i.d. processes, the estimated $\mathbf{B}$ in practice is not an exact identity matrix.

III.B.2 T-MSBL: An Efficient Algorithm Processing in the Original Problem Space

The proposed T-SBL algorithm has excellent performance in terms of recovery performance. But it is not fast because it learns the parameters in a higher dimensional space instead of the original problem space. For example, the dictionary matrix is of the size $ML \times NL$ in the bSBL framework, while it is only of the size $M \times N$ in the original MMV model. Interestingly, the MSBL developed for i.i.d. sources has complexity $\mathcal{O}(M^2N)$ and does not exhibit this drawback [154]. Motivated by this, we make a reasonable approximation and back-map T-SBL to the original space [7].

For convenience, we first list the MSBL algorithm derived in [154]:

$$\begin{aligned} \boldsymbol{\Xi}_x &= \left(\boldsymbol{\Gamma}^{-1} + \frac{1}{\lambda}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1} && \text{(III.24)} \\ \mathbf{X} &= \boldsymbol{\Gamma}\boldsymbol{\Phi}^T\left(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\right)^{-1}\mathbf{Y} && \text{(III.25)} \\ \gamma_i &= \frac{1}{L}\|\mathbf{X}_{i\cdot}\|_2^2 + (\boldsymbol{\Xi}_x)_{ii}, \quad \forall i && \text{(III.26)} \end{aligned}$$

An important observation is the lower dimension of the matrix operations involved in this algorithm. We attempt to achieve similar complexity for the T-SBL algorithm by adopting the following approximation:

$$\begin{aligned} \left(\lambda\mathbf{I}_{ML} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\right)^{-1} &= \left(\lambda\mathbf{I}_{ML} + (\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T) \otimes \mathbf{B}\right)^{-1} \\ &\approx \left(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\right)^{-1} \otimes \mathbf{B}^{-1} && \text{(III.27)} \end{aligned}$$

which is exact when $\lambda = 0$ or $\mathbf{B} = \mathbf{I}_L$. For high SNR or low correlation the approximation is quite reasonable. But experiments show that our algorithm adopting this approximation performs quite well over a broader range of conditions (see Section III.E).

Now we use the approximation to simplify the γ_i learning rule (III.18).

[7] By back-mapping, we mean we use some approximation to simplify the algorithm such that the simplified version directly operates in the parameter space of the original MMV model.

First, we consider the following term in (III.18):

$$\begin{aligned}
\frac{1}{L}\mathrm{Tr}\big(\mathbf{B}^{-1}\boldsymbol{\Sigma}_x^i\big) &= \frac{1}{L}\mathrm{Tr}\Big[\gamma_i\mathbf{I}_L - \gamma_i^2(\boldsymbol{\phi}_i^T\otimes\mathbf{I}_L)(\lambda\mathbf{I}_{ML} + \\
&\quad \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}(\boldsymbol{\phi}_i\otimes\mathbf{I}_L)\cdot\mathbf{B}\Big] \qquad \text{(III.28)}\\
&\approx \gamma_i - \frac{\gamma_i^2}{L}\mathrm{Tr}\Big[\Big(\big[\boldsymbol{\phi}_i^T\big(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\boldsymbol{\phi}_i\big]\otimes\mathbf{B}^{-1}\Big)\mathbf{B}\Big]\\
&= \gamma_i - \frac{\gamma_i^2}{L}\mathrm{Tr}\Big[\Big(\boldsymbol{\phi}_i^T\big(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\boldsymbol{\phi}_i\Big)\mathbf{I}_L\Big]\\
&= \gamma_i - \gamma_i^2\boldsymbol{\phi}_i^T\big(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\boldsymbol{\phi}_i\\
&= (\boldsymbol{\Xi}_x)_{ii} \qquad \text{(III.29)}
\end{aligned}$$

where (III.28) follows the second equation in (III.9), and $\boldsymbol{\Xi}_x$ is given in (III.24). Using the same approximation (III.27), the $\boldsymbol{\mu}_x$ in (III.18) can be expressed as

$$\begin{aligned}
\boldsymbol{\mu}_x &\approx (\boldsymbol{\Gamma}\otimes\mathbf{B})(\boldsymbol{\Phi}^T\otimes\mathbf{I})\cdot\big[\big(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\otimes\mathbf{B}^{-1}\big]\mathrm{vec}(\mathbf{Y}^T) \qquad \text{(III.30)}\\
&= \big[\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\big]\otimes\mathbf{I}\cdot\mathrm{vec}(\mathbf{Y}^T)\\
&= \mathrm{vec}\big(\mathbf{Y}^T\big(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}\big)\\
&= \mathrm{vec}(\mathbf{X}^T) \qquad \text{(III.31)}
\end{aligned}$$

where (III.43) follows (III.7) and the approximation (III.27), and $\mathbf{X}$ is given in (III.25). Therefore, based on (III.29) and (III.31), we can transform the γ_i learning rule (III.18) to the following form:

$$\gamma_i \leftarrow \frac{1}{L}\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T + (\boldsymbol{\Xi}_x)_{ii}, \quad \forall i \qquad \text{(III.32)}$$

To simplify the $\mathbf{B}$ learning rule (III.20), we note that

$$\begin{aligned}
\boldsymbol{\Sigma}_x &= \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\mathbf{D}^T(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{D}\boldsymbol{\Sigma}_0\\
&= \boldsymbol{\Gamma}\otimes\mathbf{B} - (\boldsymbol{\Gamma}\otimes\mathbf{B})(\boldsymbol{\Phi}^T\otimes\mathbf{I})(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}(\boldsymbol{\Phi}\otimes\mathbf{I})(\boldsymbol{\Gamma}\otimes\mathbf{B})\\
&\approx \boldsymbol{\Gamma}\otimes\mathbf{B} - \big[(\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)\otimes\mathbf{B}\big]\big[(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\otimes\mathbf{B}^{-1}\big]\big[(\boldsymbol{\Phi}\boldsymbol{\Gamma})\otimes\mathbf{B}\big] \qquad \text{(III.33)}\\
&= \big(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}\big)\otimes\mathbf{B}\\
&= \boldsymbol{\Xi}_x\otimes\mathbf{B}, \qquad \text{(III.34)}
\end{aligned}$$

where (III.45) uses the approximation (III.27). Using the definition (III.17), we have $\mathbf{\Sigma}_x^i = (\mathbf{\Xi}_x)_{ii}\mathbf{B}$. Therefore, the learning rule (III.20) becomes:

$$\mathbf{B} \leftarrow \Big(\frac{1}{N}\sum_{i=1}^{N}\frac{(\mathbf{\Xi}_x)_{ii}}{\gamma_i}\Big)\mathbf{B} + \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i}. \tag{III.35}$$

From the learning rule above, we can directly construct a fixed-point learning rule, given by

$$\mathbf{B} \leftarrow \frac{1}{N(1-\rho)}\sum_{i=1}^{N}\frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} \tag{III.36}$$

where $\rho = \frac{1}{N}\sum_{i=1}^{N}\gamma_i^{-1}(\mathbf{\Xi}_x)_{ii}$. To increase the robustness, however, we suggest using the rule below:

$$\widetilde{\mathbf{B}} \leftarrow \sum_{i=1}^{N}\frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} \tag{III.37}$$

$$\mathbf{B} \leftarrow \widetilde{\mathbf{B}}/\|\widetilde{\mathbf{B}}\|_{\mathcal{F}} \tag{III.38}$$

where (III.38) is to remove the ambiguity between $\mathbf{B}$ and γ_i ($\forall i$). This learning rule performs well in high SNR cases and noiseless cases [8]. However, in low or medium SNR cases (e.g. SNR $\leq$ 20dB) it is not robust due to errors from the estimated γ_i and $\mathbf{X}_{i\cdot}$. For these cases, we suggest adding a regularization item in $\widetilde{\mathbf{B}}$, namely,

$$\widetilde{\mathbf{B}} \leftarrow \sum_{i=1}^{N}\frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} + \eta\mathbf{I} \tag{III.39}$$

where η is a positive scalar. This regularized form (III.39) ensures that $\widetilde{\mathbf{B}}$ is positive definite.

[8] Note that in (III.37) when the number of distinct nonzero rows in $\mathbf{X}$ is smaller than the number of measurement vectors, the matrix $\widetilde{\mathbf{B}}$ is not invertible. But this case is rarely encountered in practical problems, since in practice the number of measurement vectors is generally small, as we explained previously. The presence of noise in practical problems also requires the use of the regularized form (III.39), which is always invertible.

Similarly, we simplify the λ learning rule (III.23) as follows:

$$\begin{aligned}
\lambda \;\leftarrow\; & \frac{\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\lambda\big[NL-\mathrm{Tr}(\boldsymbol{\Sigma}_x\boldsymbol{\Sigma}_0^{-1})\big]}{ML} \\
= \;& \frac{\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\lambda\mathrm{Tr}(\boldsymbol{\Sigma}_0\mathbf{D}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{D})}{ML} && \text{(III.40)} \\
\approx \;& \frac{1}{ML}\|\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X}\|_{\mathcal{F}}^2+\frac{\lambda}{ML}\mathrm{Tr}\Big[(\boldsymbol{\Gamma}\otimes\mathbf{B})(\boldsymbol{\Phi}^T\otimes\mathbf{I}) \\
& \cdot\big((\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\otimes\mathbf{B}^{-1}\big)(\boldsymbol{\Phi}\otimes\mathbf{I})\Big] && \text{(III.41)} \\
= \;& \frac{1}{ML}\|\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X}\|_{\mathcal{F}}^2+\frac{\lambda}{M}\mathrm{Tr}\big[\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T(\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\big] && \text{(III.42)}
\end{aligned}$$

where in (III.44) we use the first equation in (III.9), and in (III.41) we use the approximation (III.27). Empirically, we find that setting the off-diagonal elements of $\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T$ to zeros further improves the robustness of the λ learning rule in strongly noisy cases. In our experiments we will use the modified version when SNR $\leq$ 20dB.

We denote the algorithm using the learning rules (III.24), (III.25), (III.32), (III.37), (III.38) (or (III.39)), and (III.42) by **T-MSBL** (the name emphasizes the algorithm is a *temporal* extension of MSBL). Note that T-MSBL cannot be derived by modifying the cost function of MSBL.

Comparing the γ_i learning rule of T-MSBL (Eq.(III.32)) with the one of MSBL (Eq.(III.26)), we observe that the only change is the replacement of $\|\mathbf{X}_{i\cdot}\|_2^2$ with $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$, which incorporates the temporal correlation of the sources. Hence, T-MSBL has only extra computational load for calculating the matrix $\mathbf{B}$ and the item $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$ [9]. Since the matrix $\mathbf{B}$ has a small size and is positive definite and symmetric, the extra computational load is low.

Note that $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$ is the quadratic Mahalanobis distance between $\mathbf{X}_{i\cdot}$ and its mean (a vector of zeros). Later we will get more insight into this change.

[9] Here we do not compare the λ learning rules of both algorithms, since in some cases one can feed the algorithms with suitable fixed values of λ, instead of using the λ learning rules. However, the computational load of the simplified λ learning rule of T-MSBL is also not high.

III.B.3 T-MSBL-FP: A Variant of T-MSBL Based on MacKay's Fixed-Point Method

Although T-MSBL is much faster than T-SBL, it is still not fast compared to other MMV algorithms. Thus, we derive a variant of T-MSBL based on MacKay's fixed-point method [83]. This fixed-point method has been used by Tipping [132] to derive a basic SBL algorithm.

To conveniently derive learning rules, we first simplify $\mathcal{L}(\Theta)$ (III.11). First, note that

$$\begin{aligned}
\mathbf{y}^T(\boldsymbol{\Sigma}_y)^{-1}\mathbf{y} &= \mathbf{y}^T(\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{y} \\
&= \mathbf{y}^T\big[\frac{1}{\lambda}\mathbf{I}-\frac{1}{\lambda^2}\mathbf{D}(\boldsymbol{\Sigma}_0^{-1}+\frac{1}{\lambda}\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\big]\mathbf{y} \\
&= \frac{1}{\lambda}\mathbf{y}^T\big[\mathbf{y}-\mathbf{D}(\lambda\boldsymbol{\Sigma}_0^{-1}+\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{y}\big] \\
&= \frac{1}{\lambda}\mathbf{y}^T\big[\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\big] \qquad \text{(III.43)} \\
&= \frac{1}{\lambda}\big[\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\boldsymbol{\mu}_x^T\mathbf{D}^T\mathbf{y}-\boldsymbol{\mu}_x^T\mathbf{D}^T\mathbf{D}\boldsymbol{\mu}_x\big] \\
&= \frac{1}{\lambda}\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\boldsymbol{\mu}_x^T\big(\boldsymbol{\Sigma}_x^{-1}-\frac{1}{\lambda}\mathbf{D}^T\mathbf{D}\big)\boldsymbol{\mu}_x \qquad \text{(III.44)} \\
&= \frac{1}{\lambda}\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\boldsymbol{\mu}_x^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_x \qquad \text{(III.45)}
\end{aligned}$$

where (III.43) and (III.44) both used the equation (III.7), and (III.45) used the equation (III.8). Next, using the Sylvester's Determinant Theorem, we have

$$\begin{aligned}
\log|\boldsymbol{\Sigma}_y| &= \log|\lambda\mathbf{I}_{ML}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T| \\
&= \log|\lambda\mathbf{I}_{ML}|+\log\big|\mathbf{I}_{NL}+\frac{1}{\lambda}\boldsymbol{\Sigma}_0^{\frac{1}{2}}\mathbf{D}^T\mathbf{D}\boldsymbol{\Sigma}_0^{\frac{1}{2}}\big| \qquad \text{(III.46)} \\
&= \log|\lambda\mathbf{I}_{ML}|+\log|\boldsymbol{\Sigma}_0^{-1}+\frac{1}{\lambda}\mathbf{D}^T\mathbf{D}|+\log|\boldsymbol{\Sigma}_0|. \qquad \text{(III.47)}
\end{aligned}$$

Combining (III.45) and (III.47), the cost function becomes

$$\begin{aligned}
\mathcal{L}(\Theta) &= \frac{1}{\lambda}\|\mathbf{y}-\mathbf{D}\boldsymbol{\mu}_x\|_2^2+\boldsymbol{\mu}_x^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_x+\log|\lambda\mathbf{I}_{ML}| \\
&\quad +\log\big|\boldsymbol{\Sigma}_0^{-1}+\frac{1}{\lambda}\mathbf{D}^T\mathbf{D}\big|+\log|\boldsymbol{\Sigma}_0|. \qquad \text{(III.48)}
\end{aligned}$$

Now it is convenient to minimize the cost function with respect to each hyperparameter.

The derivative of $\mathcal{L}(\Theta)$ with respect to γ_i is

$$\frac{\partial \mathcal{L}}{\partial \gamma_i} = -\frac{(\boldsymbol{\mu}_x^i)^T \mathbf{B}^{-1} \boldsymbol{\mu}_x^i}{\gamma_i^2} - \frac{\mathrm{Tr}\big(\boldsymbol{\Sigma}_x^i \mathbf{B}^{-1}\big)}{\gamma_i^2} + \frac{L}{\gamma_i}$$

where $\boldsymbol{\mu}_x^i$ and $\boldsymbol{\Sigma}_x^i$ have been defined in (III.17). Letting $\frac{\partial \mathcal{L}}{\partial \gamma_i} = 0$ and following MacKay's fixed-point approach [84, 132], we have

$$\gamma_i \leftarrow \frac{(\boldsymbol{\mu}_x^i)^T \mathbf{B}^{-1} \boldsymbol{\mu}_x^i}{L - \mathrm{Tr}\big(\boldsymbol{\Sigma}_x^i \mathbf{B}^{-1}\big)/\gamma_i}, \quad i = 1, \cdots, N \tag{III.49}$$

The learning rules for $\mathbf{B}$ and λ can be derived using the EM method, which are the same as those of T-MSBL:

$$\mathbf{B} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \frac{\boldsymbol{\mu}_x^i (\boldsymbol{\mu}_x^i)^T + \boldsymbol{\Sigma}_x^i}{\gamma_i} \tag{III.50}$$

$$\lambda \leftarrow \frac{\|\mathbf{y} - \mathbf{D}\boldsymbol{\mu}_x^i\|_2^2 + \lambda\big[NL - \mathrm{Tr}(\boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_0^{-1})\big]}{ML}. \tag{III.51}$$

The learning rules (III.7), (III.8), (III.49), (III.50), and (III.51) comprise an algorithm, which, as T-SBL, does not operate in the original MMV model. Thus, we use the same approximation equation (III.27) to simplify it. Following the simplification procedure in the previous section, we obtain the simplified algorithm as follows:

$$\begin{aligned}
\boldsymbol{\Xi} &\leftarrow \big(\boldsymbol{\Gamma}^{-1} + \frac{1}{\lambda}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\big)^{-1} \\
\mathbf{X} &\leftarrow \boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T\big)^{-1}\mathbf{Y} \\
\gamma_i &\leftarrow \frac{\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T}{L(1 - \boldsymbol{\Xi}_{ii}/\gamma_i)}, \quad \forall i \\
\mathbf{B} &\leftarrow \widetilde{\mathbf{B}}/\|\widetilde{\mathbf{B}}\|_{\mathcal{F}}, \quad \text{with} \quad \widetilde{\mathbf{B}} = \sum_{i=1}^{N} \frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} + \eta\mathbf{I} \\
\lambda &\leftarrow \frac{1}{ML}\|\mathbf{Y} - \boldsymbol{\Phi}\mathbf{X}\|_{\mathcal{F}}^2 + \frac{\lambda}{M}\mathrm{Tr}\big[\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\big]
\end{aligned}$$

where $\boldsymbol{\Xi}_{ii}$ is the (i,i)-th element of $\boldsymbol{\Xi}$. We denote the algorithm by **T-MSBL-FP**. Note that the robustness of the λ learning rule in noisy environment can be improved by setting the off-diagonal elements of $\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T$ to zero, as in T-MSBL.

T-MSBL-FP is much faster than T-MSBL. But more interestingly, from its cost function (III.11) we can connect it to many well-established algorithms, providing insights to our algorithm and motivations to design new algorithms. This will be elaborated in the following.

III.C Connections to Existing Algorithms

Iterative reweighted algorithms for the MMV model can be categorized into two classes. One is the iterative reweighted ℓ_1 algorithms, which have the form

$$\mathbf{X}^{(k+1)} \leftarrow \arg\min_{\mathbf{X}} \|\mathbf{Y}-\mathbf{\Phi X}\|_{\mathcal{F}}^2 + \lambda \sum_i w_i^{(k)} \|\mathbf{X}_{i\cdot}\|_q \tag{III.52}$$

where k indicates the iteration number, and typically $q=2$ or $q=\infty$. $w_i^{(k)}$ is the weight of $\mathbf{X}_{i\cdot}$, which depends on the estimates of $\mathbf{X}$ in previous iterations. One can see the widely used Group-Lasso (for the MMV model) is its single iteration.

Another class is the iterative reweighted ℓ_2 algorithms, which have the form:

$$\mathbf{X}^{(k+1)} \leftarrow \arg\min_{\mathbf{X}} \|\mathbf{Y}-\mathbf{\Phi X}\|_{\mathcal{F}}^2 + \lambda \sum_i w_i^{(k)} (\|\mathbf{X}_{i\cdot}\|_q)^2 \tag{III.53}$$

where typically $q=2$ or $q=\infty$. When $q=2$, (III.53) has the close form:

$$\mathbf{X}^{(k+1)} \leftarrow \mathbf{W}^{(k)}\mathbf{\Phi}^T\big(\lambda\mathbf{I}+\mathbf{\Phi W}^{(k)}\mathbf{\Phi}^T\big)^{-1}\mathbf{Y} \tag{III.54}$$

where $\mathbf{W}^{(k)}$ is a diagonal weighting matrix at the k-th iteration with the i-th diagonal element being $1/w_i^{(k)}$. The M-FOCUSS algorithm [25] belongs to this class.

This section builds connections between T-MSBL and the two classes of iterative reweighted algorithms, and then suggests a strategy to improve existing iterative reweighted algorithms to incorporate temporal correlation for better performance. The effectiveness of this strategy is confirmed by two examples.

III.C.1 Connection to Iterative Reweighted ℓ_1 Algorithms

We consider to transform the cost function (III.11). Using the identity $\mathbf{y}^T(\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{y} \equiv \min_{\mathbf{x}}\left[\frac{1}{\lambda}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2+\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x}\right]$ (see Chapter II), the upper-bound of the cost function is

$$\mathfrak{L}(\mathbf{x},\gamma,\mathbf{B}) \;\; = \;\; \log|\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T| + \frac{1}{\lambda}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x}.$$

By first minimizing it over $\boldsymbol{\gamma}$ and $\mathbf{B}$ and then minimizing over $\mathbf{x}$, we have:

$$\mathbf{x} = \arg\min_{\mathbf{x}}\Big\{\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \lambda g_{\mathrm{C}}(\mathbf{x})\Big\}, \tag{III.55}$$

with the penalty $g_{\mathrm{C}}(\mathbf{x})$ given by

$$g_{\mathrm{C}}(\mathbf{x}) \triangleq \min_{\boldsymbol{\gamma}\succeq\mathbf{0},\mathbf{B}\succ\mathbf{0}}\Big\{\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T|\Big\}. \tag{III.56}$$

One may immediately realize that the problem (III.55)-(III.56) is the same as the one (II.25)-(II.26) in deriving the BSBL-ℓ_1 algorithm. Therefore, following the procedure in deriving the BSBL-ℓ_1 algorithm, we obtain

$$\mathbf{x} = \arg\min_{\mathbf{x}}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \lambda\Big[\min_{\mathbf{z}\succeq\mathbf{0},\mathbf{B}\succ\mathbf{0}}\sum_i\big(2z_i^{\frac{1}{2}}\sqrt{\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i}\big) - h^*(\mathbf{z})\Big] \tag{III.57}$$

where $\mathbf{z} \triangleq [z_1,\cdots,z_N]^T$, $h^*(\mathbf{z})$ is the conjugate concave function of $h(\boldsymbol{\gamma}) \triangleq \log|\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T|$, i.e.,

$$h^*(\mathbf{z}) = \min_{\boldsymbol{\gamma}\succeq 0}\mathbf{z}^T\boldsymbol{\gamma} - h(\boldsymbol{\gamma}), \tag{III.58}$$

and the optimal value of γ_i is given by

$$\gamma_i = z_i^{-\frac{1}{2}}\sqrt{\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i}, \quad \forall i \tag{III.59}$$

To obtain the solution $\mathbf{x}$, we need to first calculate the optimal values of $\mathbf{B}$ and z_i. The optimal value of $\mathbf{B}$ can be obtained from (III.56). Note that:

$$\frac{\partial}{\partial\mathbf{B}}[\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\lambda\mathbf{I}+\mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T|] = \sum_i\big[-\mathbf{B}^{-1}\mathbf{x}_i\mathbf{x}_i^T\mathbf{B}^{-1}/\gamma_i + \gamma_i\mathbf{D}_i^T\boldsymbol{\Sigma}_y^{-1}\mathbf{D}_i\big]$$

where $\mathbf{D}_i \triangleq \mathbf{\Phi}_i \otimes \mathbf{I}_L$ and $\mathbf{\Phi}_i$ is the i-th column of $\mathbf{\Phi}$. Setting it to zero, we have

$$\begin{aligned}
\mathbf{B}^{-1}\sum_i \frac{\mathbf{x}_i\mathbf{x}_i^T}{\gamma_i}\mathbf{B}^{-1} &= \sum_i \gamma_i \mathbf{D}_i^T \mathbf{\Sigma}_y^{-1}\mathbf{D}_i \\
&= \sum_i \gamma_i (\mathbf{\Phi}_i^T \otimes \mathbf{I})\big(\lambda \mathbf{I}_{NL} + (\mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)\otimes \mathbf{B}\big)^{-1}(\mathbf{\Phi}_i \otimes \mathbf{I}) \\
&\overset{(*)}{\approx} \sum_i \gamma_i (\mathbf{\Phi}_i^T \otimes \mathbf{I})\big[(\lambda \mathbf{I}_N + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)^{-1}\otimes \mathbf{B}^{-1}\big]\cdot(\mathbf{\Phi}_i \otimes \mathbf{I}) \\
&= \Big[\sum_i \gamma_i \mathbf{\Phi}_i^T(\lambda \mathbf{I} + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}_i\Big]\mathbf{B}^{-1}
\end{aligned}$$

where (*) used the approximation (III.27). Thus, we obtain the learning rule

$$\mathbf{B} = \frac{1}{C}\sum_{i=1}^N \frac{\mathbf{x}_i\mathbf{x}_i^T}{\gamma_i} = \frac{1}{C}\sum_{i=1}^N \frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} \tag{III.60}$$

with $C \triangleq \sum_{i=1}^N \gamma_i \mathbf{\Phi}_i^T(\lambda \mathbf{I} + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}_i$.

Using the property of conjugate functions [13, Chapter 3.3], from (III.58) we can directly obtain the optimal z_i as follows $z_i = \frac{\partial \log|\lambda \mathbf{I} + \mathbf{D}\mathbf{\Sigma}_0\mathbf{D}^T|}{\partial \gamma_i} = \mathrm{Tr}\big[\mathbf{B}\mathbf{D}_i^T\big(\lambda \mathbf{I} + \mathbf{D}\mathbf{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{D}_i\big]$. Hence,

$$\begin{aligned}
z_i^{\frac{1}{2}} &= \Big(\mathrm{Tr}\big[\mathbf{B}\mathbf{D}_i^T\big(\lambda \mathbf{I} + \mathbf{D}\mathbf{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{D}_i\big]\Big)^{\frac{1}{2}} \\
&\approx \sqrt{L\mathbf{\Phi}_i^T(\lambda \mathbf{I} + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}_i},
\end{aligned} \tag{III.61}$$

where we used the approximation (III.27) again.

Based on the above development, we see that the optimal values of $\mathbf{B}$ and z_i depend on $\mathbf{X}$ itself. Thus the whole learning procedure is an iterative algorithm. In the k-th iteration, once having used the updating rules (III.59) (III.60) and (III.61) to obtain $\mathbf{B}^{(k)}$ and the weight $w_i^{(k)} \triangleq 2z_i^{1/2}$, we only need to solve the following optimization problem:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \sum_i w_i^{(k)} \sqrt{\mathbf{x}_i^T(\mathbf{B}^{(k)})^{-1}\mathbf{x}_i}.$$

Hence, from the cost function (III.11) we obtain the iterative reweighted ℓ_1 algorithm,

$$\mathbf{X}^{(k+1)} \leftarrow \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{\Phi}\mathbf{X}\|_{\mathcal{F}}^2 + \lambda \sum_i w_i \sqrt{\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T} \tag{III.62}$$

with the weights given by

$$\begin{aligned} w_i &\leftarrow 2\sqrt{L\boldsymbol{\Phi}_i^T(\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}_i}, \quad \forall i \\ \gamma_i &\leftarrow \frac{\sqrt{\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T}}{\sqrt{L\boldsymbol{\Phi}_i^T(\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}_i}}, \quad \forall i \\ \mathbf{B} &\leftarrow \frac{1}{C}\sum_{i=1}^{N}\frac{\mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}}{\gamma_i} \end{aligned}$$

with $C \triangleq \sum_{i=1}^{N}\gamma_i\boldsymbol{\Phi}_i^T(\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}_i$.

Now we draw the connection to existing algorithms. From (III.62) we can see our penalty is a temporal-correlation-aware penalty, and the temporal correlation structure is adaptively learned from data. This is entirely different from the penalties used in Group-Lasso type algorithms (for the MMV model) and most iterative reweighted ℓ_1 algorithms (III.52), which are blind to the temporal correlation.

In fact, the matrix $\mathbf{B}$ in our penalty can be viewed as a data-adaptive kernel. This is different from the non-adaptive kernels used in some existing mixed $\ell_{2,1}$-norm penalties [157, 176, 163], which generally need users to design kernels according to some a priori knowledge or by cross-validation. Note that the data-adaptive kernel is advantageous over the user-defined kernels, because in some applications such as our application, a priori knowledge may not be available. Also, user-designed kernels cannot accurately capture the correlation structure of data, which is a serious problem for regression.

Clearly, the algorithm (III.62) is an MMV-model based iterative reweighted ℓ_1 minimization algorithm, since its weights w_i depends on the estimate of $\mathbf{X}$ in previous iterations. In contrast, the Group-Lasso type algorithms are just a single iteration of it (with $\mathbf{B} = \mathbf{I}$). It is known that iterative reweighted algorithms have better performance than their non-iterative-reweighted counterparts and can provide more sparse solutions [17, 149].

The above observations motivate us to improve existing Group-Lasso type algorithms and iterative reweighted ℓ_1 algorithms by adaptively learning the cor-

relation structure of data. The details are given in the next subsection.

III.C.2 Improve Existing Iterative Reweighted ℓ_1 Algorithms by Exploiting Temporal Correlation

The connection between (III.62) and the canonical iterative reweighted ℓ_1 algorithms (III.52) suggests a strategy to incorporate correlation structure into the latter by replacing $\|\mathbf{X}_{i\cdot}\|_q$ with the Mahalanobis-distance type measure $\sqrt{\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T}$ while the kernel $\mathbf{B}$ is adaptively learned from data.

Below we give an example to show how to do this. A canonical iterative reweighted ℓ_1 is given below

$$\begin{aligned}\mathbf{X}^{(k+1)} &= \arg\min_{\mathbf{X}} \|\mathbf{Y}-\mathbf{\Phi}\mathbf{X}\|_{\mathcal{F}}^2 + \lambda\sum_i w_i^{(k)}\|\mathbf{X}_{i\cdot}\|_2 \\ w_i^{(k)} &= \left(\|\mathbf{X}_{i\cdot}^{(k)}\|_2+\epsilon\right)^{-1}\end{aligned}$$

where ϵ is a constant. This algorithm is an MMV form of the one in [17]. We modify its weights as follows

$$w_i^{(k)} = \left(\sqrt{\mathbf{X}_{i\cdot}^{(k)}(\mathbf{B}^{(k)})^{-1}(\mathbf{X}_{i\cdot}^{(k)})^T}+\epsilon\right)^{-1}, \tag{III.63}$$

where $\mathbf{B}^{(k)}$ can be estimated from the estimate of $\mathbf{X}$ in the previous iteration (the estimation method is similar to the one in BSBL-ℓ_1).

To see the improvement using the new weight (III.63), a noiseless experiment was carried out. In the experiment the Gaussian random matrix $\mathbf{\Phi}$ was of the size 25×125. The number of measurement vectors was 4. The number of nonzero rows of $\mathbf{X}$ was 12. Each nonzero row was generated as an AR(1) process with the AR coefficient being 0.9. The ℓ_2-norm of each nonzero row was uniformed distributed in $[0.3, 1]$. To clearly see the advantage of the new weights, we set $\mathbf{B}^{(k)}$ $(\forall k)$ to be the true value, i.e. $\mathbf{B}^{(k)} = \text{Toeplitz}([1, 0.9, 0.9^2, 0.9^3])$.

In addition to the comparison between the canonical iterative reweighted ℓ_1 algorithm and the modified one, we also compared the iterative reweighted ℓ_1

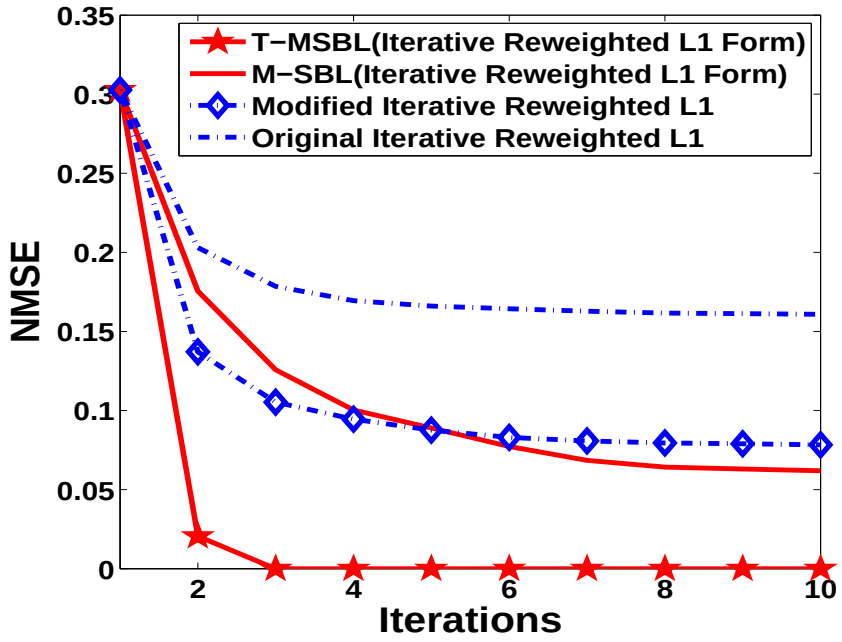


Figure III.1 Performance comparison between the two iterative reweighted ℓ_1 algorithms and their improved counterparts.

form of T-MSBL (III.62) with the iterative reweighted ℓ_1 form of M-SBL in [149]. The two only differs in $\mathbf{B}$. In the latter, $\mathbf{B}^{(k)} = \mathbf{I}(\forall k)$.

The averaged results over 100 trials are shown in Figure III.1, where we can see the modified iterative reweighted ℓ_1 had improved performance, compared to the original one. Besides, the iterative reweighted ℓ_1 form of T-MSBL (III.62) had better performance than the iterative reweighted ℓ_1 form of M-SBL. These results imply that the strategy to improve existing iterative reweighted ℓ_1 by incorporating temporal correlation into the weights (and/or the penalties) is promising. However, more theoretical studies are required along this line.

III.C.3 Connection to Iterative Reweighted ℓ_2 Algorithms

Now we connect T-MSBL (and T-MSBL-FP) to the iterative reweighted ℓ_2 algorithms (III.53).

As in Section III.C.1, we can transform the cost function (III.11) to

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda g_{\mathrm{C}}(\mathbf{x}), \tag{III.64}$$

where the penalty $g_{\mathrm{C}}(\mathbf{x})$ is defined by

$$g_{\mathrm{C}}(\mathbf{x}) \triangleq \min_{\boldsymbol{\gamma}\succeq\mathbf{0},\mathbf{B}\succ\mathbf{0}} \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T|. \tag{III.65}$$

Note that

$$\begin{aligned} g_{\mathrm{C}}(\mathbf{x}) &\leq \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T| \\ &= \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\boldsymbol{\Sigma}_0| + \log|\frac{1}{\lambda}\mathbf{D}^T\mathbf{D} + \boldsymbol{\Sigma}_0^{-1}| + ML\log\lambda \\ &\leq \mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \log|\boldsymbol{\Sigma}_0| + \mathbf{z}^T\boldsymbol{\gamma}^{-1} - f^*(\mathbf{z}) + ML\log\lambda \end{aligned}$$

where in the last inequality we have used the conjugate relation

$$f(\boldsymbol{\gamma}^{-1}) \triangleq \log\Big|\frac{1}{\lambda}\mathbf{D}^T\mathbf{D} + \boldsymbol{\Sigma}_0^{-1}\Big| = \min_{\mathbf{z}\succeq 0} \mathbf{z}^T\boldsymbol{\gamma}^{-1} - f^*(\mathbf{z}). \tag{III.66}$$

Here we denote $\boldsymbol{\gamma}^{-1} \triangleq [\gamma_1^{-1}, \cdots, \gamma_N^{-1}]^T$, $\mathbf{z} \triangleq [z_1, \cdots, z_N]^T$, and $f^*(\mathbf{z})$ is concave conjugate of $f(\boldsymbol{\gamma}^{-1})$. Finally, reminding of $\boldsymbol{\Sigma}_0 = \boldsymbol{\Gamma} \otimes \mathbf{B}$, we have

$$g_{\mathrm{C}}(\mathbf{x}) \leq ML\log\lambda - f^*(\mathbf{z}) + N\log|\mathbf{B}| + \sum_{i=1}^{N}\Big[\frac{\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i + z_i}{\gamma_i} + L\log\gamma_i\Big]. \tag{III.67}$$

Therefore, to solve the problem (III.64) with (III.67), we can perform the coordinate descent method over $\mathbf{x}, \mathbf{B}, \mathbf{z}$ and $\boldsymbol{\gamma}$, i.e,

$$\min_{\mathbf{x},\mathbf{B},\mathbf{z}\succeq 0,\boldsymbol{\gamma}\succeq 0} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\Big[\sum_{i=1}^{N}\Big(\frac{\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i + z_i}{\gamma_i} + L\log\gamma_i\Big) + N\log|\mathbf{B}| - f^*(\mathbf{z})\Big]. \tag{III.68}$$

Compared to the iterative reweighted ℓ_2 framework (III.53), $1/\gamma_i$ can be seen as the weight for the corresponding $\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i$. But instead of applying ℓ_q norm on $\mathbf{x}_i$ (i.e. the i-th row of $\mathbf{X}$) as done in existing iterative reweighted ℓ_2 algorithms, our algorithm computes $\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i$, i.e. the quadratic Mahalanobis-distance measure of $\mathbf{x}_i$.

By minimizing (III.68) over $\mathbf{x}$, the updating rule for $\mathbf{x}$ is given by

$$\mathbf{x}^{(k+1)} = \boldsymbol{\Sigma}_0\mathbf{D}^T(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T)^{-1}\mathbf{y}. \tag{III.69}$$

Similar to the previous section, using the conjugate property of (III.66), the optimal $\mathbf{z}$ is directly given by

$$\begin{aligned} z_i &= \frac{\partial \log|\frac{1}{\lambda}\mathbf{D}^T\mathbf{D} + \boldsymbol{\Sigma}_0^{-1}|}{\partial(\gamma_i^{-1})} \\ &= L\gamma_i - \gamma_i^2 \mathrm{Tr}\Big[\mathbf{B}\mathbf{D}_i^T\big(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\big)^{-1}\mathbf{D}_i\Big],\ \forall i \end{aligned} \tag{III.70}$$

From (III.68) the optimal γ_i for fixed $\mathbf{x}, \mathbf{z}, \mathbf{B}$ is given by $\gamma_i = \frac{1}{L}[\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i + z_i]$. Substituting Eq.(III.70) into it, we have

$$\gamma_i^{(k+1)} = \frac{\mathbf{x}_i^T\mathbf{B}^{-1}\mathbf{x}_i}{L} + \gamma_i - \frac{\gamma_i^2}{L}\mathrm{Tr}\Big[\mathbf{B}\mathbf{D}_i^T\big(\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\big)^{-1}\mathbf{D}_i\Big],\ \forall i \tag{III.71}$$

By minimizing (III.68) over $\mathbf{B}$, the updating rule for $\mathbf{B}$ is given by

$$\mathbf{B}^{(k+1)} = \overline{\mathbf{B}}/\|\overline{\mathbf{B}}\|_{\mathcal{F}}, \quad \text{with} \quad \overline{\mathbf{B}} = \sum_{i=1}^{N}\frac{\mathbf{x}_i\mathbf{x}_i^T}{\gamma_i}. \tag{III.72}$$

The updating rules (III.69) (III.71) and (III.72) are our iterative reweighted ℓ_2 algorithm minimizing the penalty based on quadratic Mahalanobis distance of $\mathbf{x}_i$.

Similar to the back-mapping from T-SBL to T-MSBL, we use the approximation formula (III.27) to derive a simplified version.

Using the approximation (III.27), the updating rule (III.69) can be transformed to

$$\mathbf{X}^{(k+1)} = \mathbf{W}\boldsymbol{\Phi}^T\big(\lambda\mathbf{I} + \boldsymbol{\Phi}\mathbf{W}\boldsymbol{\Phi}^T\big)^{-1}\mathbf{Y}, \tag{III.73}$$

where $\mathbf{W} \triangleq \mathrm{diag}([1/w_1, \cdots, 1/w_N])$ with $w_i \triangleq 1/\gamma_i$. Using the same approximation, the last term in (III.71) becomes

$$\begin{aligned} &\mathrm{Tr}\Big[\mathbf{B}\mathbf{D}_i^T\big(\lambda\mathbf{I}_{ML} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T\big)^{-1}\mathbf{D}_i\Big] \\ &\approx \mathrm{Tr}\Big[\mathbf{B}(\boldsymbol{\Phi}_i^T \otimes \mathbf{I})\big[(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\mathbf{W}\boldsymbol{\Phi}^T)^{-1} \otimes \mathbf{B}^{-1}\big](\boldsymbol{\Phi}_i \otimes \mathbf{I})\Big] \\ &= L\boldsymbol{\Phi}_i^T(\lambda\mathbf{I}_M + \boldsymbol{\Phi}\mathbf{W}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}_i. \end{aligned}$$

Therefore, from the updating rule of γ_i in (III.71) we have

$$w_i^{(k+1)} = \Big[\frac{1}{L}\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T + \{(\mathbf{W}^{-1} + \frac{1}{\lambda}\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\}_{ii}\Big]^{-1}. \tag{III.74}$$

Accordingly, the updating rule for $\mathbf{B}$ becomes

$$\mathbf{B}^{(k+1)} = \overline{\mathbf{B}}/\|\overline{\mathbf{B}}\|_{\mathcal{F}}, \quad \text{with} \quad \overline{\mathbf{B}} = \sum_{i=1}^{N} w_i \mathbf{X}_{i\cdot}^T\mathbf{X}_{i\cdot}. \tag{III.75}$$

To estimate the regularization parameter λ, many methods have been proposed, such as those widely used for iterative reweighted ℓ_2 algorithms. But we can follow the EM method in the development of T-MSBL and use the approximation (III.27) to derive the following rule:

$$\lambda^{(k+1)} = \frac{1}{ML}\|\mathbf{Y} - \mathbf{\Phi}\mathbf{X}\|_{\mathcal{F}}^2 + \frac{\lambda}{M}\mathrm{Tr}\big[\mathbf{\Phi}\mathbf{W}\mathbf{\Phi}^T(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{W}\mathbf{\Phi}^T)^{-1}\big]. \tag{III.76}$$

The updating rules (III.73) (III.74) (III.75) and (III.76) compose the simplified version of the iterative reweighted ℓ_2 algorithm.

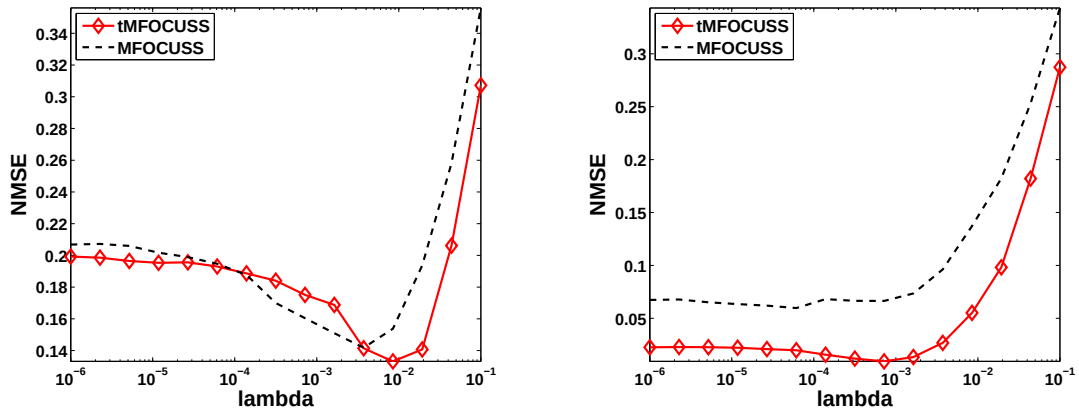
It is interesting to see that in this iterative reweighted ℓ_2 form, the weight (III.74) is completely different from the weight in the iterative reweighted ℓ_1 form.

Another observation is that the matrix $\mathbf{B}$ affects the solution via the weight (III.74). When $\mathbf{B}^{(k)} = \mathbf{I}(\forall k)$, i.e., ignoring the temporal correlation, the iterative reweighted ℓ_2 form reduces to the ℓ_2 form of M-SBL [149]. In other words, to improve the iterative reweighted ℓ_2 form of M-SBL, one can just replace the $\|\mathbf{X}_{i\cdot}\|_{\mathcal{F}}^2$ in the weight with the Mahalanobis-distance measure $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$.

III.C.4 Improve Existing Iterative Reweighted ℓ_2 Algorithms by Exploiting Temporal Correlation

Motivated by the above observation, here we give an example to show how to improve existing iterative reweighted ℓ_2 algorithms by exploiting the temporal correlation.

The regularized M-FOCUSS [25] is a typical iterative reweighted ℓ_2 algorithm, which solves a reweighted ℓ_2 minimization with weights $w_i^{(k)} = (\|\mathbf{X}_{i\cdot}^{(k)}\|_2^2)^{p/2-1}$



(a) 15dB (b) 25dB

Figure III.2 Performance comparison of tMFOCUSS and M-FOCUSS at different SNR. Each nonzero row of $\mathbf{X}$ was generated as an AR(1) process with the AR coefficient 0.9.

in each iteration. The algorithm is given by

$$\mathbf{X}^{(k+1)} = \mathbf{W}^{(k)}\mathbf{\Phi}^T\left(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{W}^{(k)}\mathbf{\Phi}^T\right)^{-1}\mathbf{Y} \quad \text{(III.77)}$$

$$\mathbf{W}^{(k)} = \text{diag}\{[1/w_1^{(k)}, \cdots, 1/w_M^{(k)}]\}$$

$$w_i^{(k)} = \left(\|\mathbf{X}_{i\cdot}^{(k)}\|_2^2\right)^{p/2-1},\ p \in [0,2], \forall i \quad \text{(III.78)}$$

We can modify the algorithm by changing the weight (III.78) to the following one:

$$w_i^{(k)} = \left(\mathbf{X}_{i\cdot}^{(k)}(\mathbf{B}^{(k)})^{-1}(\mathbf{X}_{i\cdot}^{(k)})^T\right)^{p/2-1},\ p \in [0,2], \forall i \quad \text{(III.79)}$$

The matrix $\mathbf{B}$ can be calculated using the learning rule (III.75). We denote the modified algorithm by **tMFOCUSS**.

To show the improvement, we carried out the following experiment. In this experiment the matrix Φ was a random Gaussian matrix with the size 50×200. The number of measurement vectors was 4. The number of nonzero rows of $\mathbf{X}$ was 20. Each nonzero row was generated as an AR(1) process with the AR coefficient being 0.9. We considered two SNR cases: one was 15 dB and the other was 25 dB. The original M-FOCUSS and the tMFOCUSS were performed [10]. Their regularization

[10] Matlab codes can be downloaded at: http://dsp.ucsd.edu/~zhilin/tMFOCUSS_code.zip .

parameter λ was swept from 10^{-6} to 10^{-1}, and their performance changing with different values of λ was plotted (in this way we can remove the disturbance from non-optimal values of λ).

The results averaged over 40 trials are shown in Figure III.2, where we can see that tMFOCUSS outperformed the original M-FOCUSS especially in the higher SNR case. It is worthy noticing that tMFOCUSS is simply obtained by replacing $\|\mathbf{X}_{i\cdot}\|_{\mathcal{F}}^2$ in the weight of M-FOCUSS with the $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$.

Other examples on the modification of existing iterative reweighted ℓ_2 algorithms can be found in [173].

III.D Analysis of Global Minimum and Local Minima

Due to the equivalence of the original MMV model (III.1) and the transformed block sparsity model (III.2), in the following discussions we use (III.1) or (III.2) interchangeably and per convenience.

Throughout our analysis, the true source matrix is denoted by $\mathbf{X}_{\text{gen}}$, which is the sparsest solution among all the possible solutions. The number of nonzero rows in $\mathbf{X}_{\text{gen}}$ is denoted by K_0. We assume that $\mathbf{X}_{\text{gen}}$ is full column-rank, the dictionary matrix $\mathbf{\Phi}$ satisfies the URP condition [57], and the matrix $\mathbf{B}$ (or $\mathbf{B}_i, \forall i$) and its estimate are positive definite.

III.D.1 Analysis of the Global Minimum

We have the following result on the global minimum of the cost function (III.11) [11]:

Theorem 1. *In the limit as $\lambda \to 0$, assuming $K_0 < (M+L)/2$, for the cost function (III.11) the unique global minimum $\widehat{\boldsymbol{\gamma}} \triangleq [\widehat{\gamma}_1, \cdots, \widehat{\gamma}_M]$ produces a source estimate $\widehat{\mathbf{X}}$ that equals to $\mathbf{X}_{\text{gen}}$ irrespective of the estimated $\widehat{\mathbf{B}}_i, \forall i$, where $\widehat{\mathbf{X}}$ is obtained from $\text{vec}(\widehat{\mathbf{X}}^T) = \widehat{\mathbf{x}}$ and $\widehat{\mathbf{x}}$ is computed using Eq.(III.10).*

[11] For convenience, in this theorem we consider the cost function with $\mathbf{\Sigma}_0$ given by (III.5), i.e. the one before we use our strategy to avoid the overfitting.

The proof is given in the Appendix of this chapter.

If we change the condition $K_0 < (M+L)/2$ to $K_0 < M$, then we have the conclusion that the source estimate $\widehat{\mathbf{X}}$ equals to $\mathbf{X}_{\text{gen}}$ with probability 1, irrespective of $\widehat{\mathbf{B}}_i\,(\forall i)$. This is due to the result in [41] that if $K_0 < M$ the above conclusion still holds for all $\mathbf{X}$ except on a set with zero measure.

Note that $\widehat{\boldsymbol{\gamma}}$ is a function of the estimated $\widehat{\mathbf{B}}_i\ (\forall i)$. However, the theorem implies that even when the estimated $\widehat{\mathbf{B}}_i$ is different from the true $\mathbf{B}_i$, the estimated sources are the true sources at the global minimum of the cost function. As a reminder, in deriving our algorithms, we assumed $\mathbf{B}_i = \mathbf{B}\ (\forall i)$ to avoid overfitting. The theorem ensures our algorithms using this strategy also have the global minimum property. Also, the theorem explains why MSBL has the ability to exactly recover true sources in noiseless cases even when sources are temporally correlated. But we hasten to add that this does not mean $\mathbf{B}$ is not important for the performance of the algorithms. For instance, MSBL is more frequently attracted to local minima than our proposed algorithms, as experiments show later.

III.D.2 Analysis of the Local Minima

Now we discuss the local minimum property of the cost function $\mathcal{L}$ in (III.11) with respect to $\boldsymbol{\gamma} \triangleq [\gamma_1, \cdots, \gamma_N]$, in which $\boldsymbol{\Sigma}_0 = \boldsymbol{\Gamma} \otimes \mathbf{B}$ for fixed $\mathbf{B}$. Before presenting our results, we provide two lemmas needed to prove the results.

Lemma 1. $\log|\boldsymbol{\Sigma}_y| \triangleq \log|\lambda\mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0\mathbf{D}^T|$ *is concave with respect to* $\boldsymbol{\gamma}$.

This can be shown using the composition property of concave functions [13].

Lemma 2. $\mathbf{y}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{y}$ *equals a constant* C *when* $\boldsymbol{\gamma}$ *satisfies the linear constraints*

$$\mathbf{A}\cdot(\boldsymbol{\gamma}\otimes\mathbf{1}_L) = \mathbf{b} \tag{III.80}$$

with

$$\mathbf{b} \triangleq \mathbf{y} - \lambda\mathbf{u} \tag{III.81}$$

$$\mathbf{A} \triangleq (\boldsymbol{\Phi}\otimes\mathbf{B})\text{diag}(\mathbf{D}^T\mathbf{u}) \tag{III.82}$$

where $\mathbf{A}$ *is full row rank,* $\mathbf{1}_L$ *is an* $L \times 1$ *vector of ones, and* $\mathbf{u}$ *is any fixed vector such that* $\mathbf{y}^T\mathbf{u} = C$.

The proof is given in the Appendix of this chapter.

According to the definition of basic feasible solution (BFS) [81], we know that if $\boldsymbol{\gamma}$ satisfies Eq.(III.80), then it is a BFS to (III.80) if $\|\boldsymbol{\gamma}\|_0 \leq ML$, or a degenerate BFS to (III.80) if $\|\boldsymbol{\gamma}\|_0 < ML$. Now we give the following result:

Theorem 2. *Every local minimum of the cost function* $\mathcal{L}$ *with respect to* $\boldsymbol{\gamma}$ *is achieved at a solution with* $\|\widehat{\boldsymbol{\gamma}}\|_0 \leq ML$*, regardless of the values of* λ *and* $\mathbf{B}$.

The proof is given in the Appendix of this chapter.

Admittedly, the bound on the local minima $\|\widehat{\boldsymbol{\gamma}}\|_0$ is loose, and it is not meaningful when $ML > N$. However, it is empirically found that $\|\widehat{\boldsymbol{\gamma}}\|_0$ is very smaller than ML, typically smaller than M.

Now we calculate the local minima of the cost function $\mathcal{L}$. The result can provide some insights to the role of $\mathbf{B}$. Particularly, we are more interested in the local minima satisfying $\|\widehat{\boldsymbol{\gamma}}\|_0 \leq M$, since the global minimum satisfies $\|\widehat{\boldsymbol{\gamma}}\|_0 < M$. For these local minima, we have the following result:

Lemma 3. *In noiseless cases* ($\lambda \to 0$)*, for every local minimum of* $\mathcal{L}$ *that satisfies* $\|\widehat{\boldsymbol{\gamma}}\|_0 \triangleq K \leq M$*, its* i*-th nonzero element is given by* $\widehat{\gamma}_{(i)} = \frac{1}{L}\widetilde{\mathbf{X}}_{i\cdot}\mathbf{B}^{-1}\widetilde{\mathbf{X}}_{i\cdot}^T$ $(i = 1, \cdots, K)$*, where* $\widetilde{\mathbf{X}}_{i\cdot}$ *is the* i*-th nonzero row of* $\widehat{\mathbf{X}}$ *and* $\widehat{\mathbf{X}}$ *is the basic feasible solution to* $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X}$.

The proof is given in the Appendix of this chapter.

From this lemma we immediately have the closed form of the global minimum.

$\mathbf{B}$ actually plays a role of temporally whitening the sources during the learning of $\boldsymbol{\gamma}$. To see this, assume all the sources have the same correlation structure, i.e. share the same matrix $\mathbf{B}$. Let $\mathbf{Z}_{i\cdot} \triangleq \widetilde{\mathbf{X}}_{i\cdot}\mathbf{B}^{-1/2}$. From Lemma 3, at the global minimum we have $\widehat{\gamma}_{(i)} = \frac{1}{L}\mathbf{Z}_{i\cdot}\mathbf{Z}_{i\cdot}^T$ $(i = 1, \cdots, K_0)$. On the other hand, in the case

of i.i.d. sources, at the global minimum we have $\widehat{\gamma}_{(i)} = \frac{1}{L}\widetilde{\mathbf{X}}_{i\cdot}\widetilde{\mathbf{X}}_{i\cdot}^T\,(i = 1, \cdots, K_0)$. So the results for the two cases have the same form. Since $E\{\mathbf{Z}_{i\cdot}^T\mathbf{Z}_{i\cdot}\} = \gamma_i\mathbf{I}$, we can see in the learning of $\boldsymbol{\gamma}$, $\mathbf{B}$ plays the role of whitening each source. This gives us a motivation to facilitate estimation procedures in solving the spatiotemporal sparse model (I.9), as shown in Chapter IV.

III.E Simulations

Extensive computer simulations have been conducted and a few representative and informative results are presented. All the simulations consisted of 1000 independent trials. In each trial a dictionary matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ was created with columns uniformly drawn from the surface of a unit hypersphere (except the last simulation), as advocated by Donoho et al [35]. And the source matrix $\mathbf{X}_{\text{gen}} \in \mathbb{R}^{N\times L}$ was randomly generated with K nonzero rows (i.e. sources). In each trial the indexes of the sources were randomly chosen. In most experiments (except to the experiment in Section III.E.4) each source was generated as AR(1) process. Thus the AR coefficient of the i-th source, denoted by β_i, indicated its temporal correlation. As done in [136, 8], for noiseless cases, the ℓ_2 norm of each source was rescaled to be uniformly distributed between 1/3 and 1; for noisy cases, rescaled to be unit norm. Finally, the measurement matrix $\mathbf{Y}$ was constructed by $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X}_{\text{gen}} + \mathbf{V}$ where $\mathbf{V}$ was a zero-mean homoscedastic Gaussian noise matrix with variance adjusted to have a desired value of SNR, which is defined by $\text{SNR(dB)} \triangleq 20\log_{10}(\|\boldsymbol{\Phi}\mathbf{X}_{\text{gen}}\|_{\mathcal{F}}/\|\mathbf{V}\|_{\mathcal{F}})$.

We used two performance measures. One was the *Failure Rate* defined in [154], which indicated the percentage of failed trials in the total trials. In noiseless cases, a failed trial was recognized if the indexes of estimated sources were not the same as the true indexes. In noisy cases, since any algorithm cannot recover $\mathbf{X}_{\text{gen}}$ exactly in these cases, a failed trial was recognized if the indexes of estimated sources with the K largest ℓ_2 norms were not the same as the true indexes. In

noisy cases, the *mean square error* (MSE) was also used as a performance measure, defined by $\|\widehat{\mathbf{X}} - \mathbf{X}_{\text{gen}}\|_{\mathcal{F}}^2 / \|\mathbf{X}_{\text{gen}}\|_{\mathcal{F}}^2$, where $\widehat{\mathbf{X}}$ was the estimated source matrix.

In our experiments we compared our T-SBL and T-MSBL with the following algorithms:

- MSBL, proposed in [154] [12];
- MFOCUSS, the regularized M-FOCUSS proposed in [25]. In all the experiments, we set its p-norm $p = 0.8$, as suggested by the authors [13];
- SOB-MFOCUSS, a smoothness constrained M-FOCUSS proposed in [163]. In all the experiments, we set its p-norm $p = 0.8$. For its smoothness matrix, we chose the identity matrix when $L \leq 2$, and a second-order smoothness matrix when $L \geq 3$, as suggested by the authors. Since in our experiments L is small, no overlap blocks were used [14];
- ISL0, an improved smooth ℓ_0 algorithm for the MMV model which was proposed in [66]. The regularization parameters were chosen according to the authors' suggestions [15];
- Reweighted ℓ_1/ℓ_2, an iterative reweighted ℓ_1/ℓ_2 algorithm suggested in [149]. It is an MMV extension of the iterative reweighted ℓ_1 algorithm [17] via the mixed ℓ_1/ℓ_2 norm. The algorithm is given by

  1. Set the iteration count k to zero and $w_i^{(0)} = 1\,(i = 1, \cdots, N)$
  2. Solve the weighted MMV ℓ_1 minimization problem

$$\mathbf{X}^{(k)} = \arg\min \sum_{i=1}^{N} w_i^{(k)} \|\mathbf{X}_{i\cdot}\|_2 \quad \text{s.t. } \mathbf{Y} = \mathbf{\Phi X}$$

[12] The MATLAB code was downloaded at http://dsp.ucsd.edu/~zhilin/MSBL_code.zip .

[13] The MATLAB code was downloaded at http://dsp.ucsd.edu/~zhilin/MFOCUSS.m .

[14] The MATLAB code was provided by the first author of [163] in personal communication. In the code the second-order smoothness matrix $\mathbf{S}$ was defined as (in MATLAB notations): $\mathbf{S} = \text{eye}(L) - 0.25 * (\text{diag}(\mathbf{e}(1:L-1), -1) + \text{diag}(\mathbf{e}(1:L-1), 1) + (\text{diag}(\mathbf{e}(1:L-2), -2) + \text{diag}(\mathbf{e}(1:L-2), 2)))$, where $\mathbf{e}$ is an $L \times 1$ vector with ones.

[15] The MATLAB code was provided by the first author of [66] in personal communication.

3. Update the weights for each $i = 1, \cdots, N$

$$w_i^{(k+1)} = \frac{1}{\|\mathbf{X}_{i\cdot}^{(k)}\|_2 + \epsilon^{(k)}}$$

where $\epsilon^{(k)}$ is adaptively adjusted as in [17];

4. Terminate on convergence or when k attains a specified maximum number of iterations $k_{\max}$. Otherwise, increment k and go to Step 2).

For noisy cases, Step 2) is modified to

$$\mathbf{X}^{(k)} = \arg\min \sum_{i=1}^{N} w_i^{(k)} \|\mathbf{X}_{i\cdot}\|_2 \quad \text{s.t. } \|\mathbf{Y} - \mathbf{\Phi X}\|_{\mathcal{F}} \leq \delta$$

Throughout our experiments, $k_{\max} = 5$. We implemented it using the CVX optimization toolbox [16].

In noisy cases, we chose the optimal values for the regularization parameter λ in MFOCUSS and the parameter δ in Reweighted ℓ_1/ℓ_2 by exhaustive search. Practically, we used a set of candidate parameter values and for each value we ran an algorithm for 50 trials, and then picked up the one which gave the smallest averaged failure rate. By comparing enough number of candidate values we could ensure a nearly optimal value of the regularization parameter for this algorithm. For T-MSBL, T-SBL and MSBL, we fixed $\lambda = 10^{-9}$ for noiseless cases, and used their λ learning rules for noisy cases. Besides, for T-MSBL we chose the learning rule (III.39) with $\eta = 2$ to estimate $\mathbf{B}$ when SNR $\leq$ 15dB.

For reproducibility, the experiment codes can be downloaded at http://dsp.ucsd.edu/~zhilin/TSBL_code.zip .

III.E.1 Benefit from Multiple Measurement Vectors at Different Temporal Correlation Levels

In this simulation we study how algorithms benefit from multiple measurement vectors and how the benefit is discounted by the temporal correlation of

[16] The toolbox was downloaded at: http://cvxr.com/cvx/.

sources. The dictionary matrix $\mathbf{\Phi}$ was of the size 25×125 and the number of sources $K = 12$. The number of measurement vectors L varied from 1 to 4. No noise was added. All the sources were AR(1) processes with the common AR coefficient β, such that we could easily observe the relationship between temporal correlation and algorithm performance. Note that for small L, modeling sources as AR(1) processes, instead of AR(p) processes with $p > 1$, is sufficient to cover wide ranges of temporal structure. We compared algorithms at six different temporal correlation levels, i.e. $\beta = -0.9$, -0.5, 0, 0.5, 0.9, 0.99.

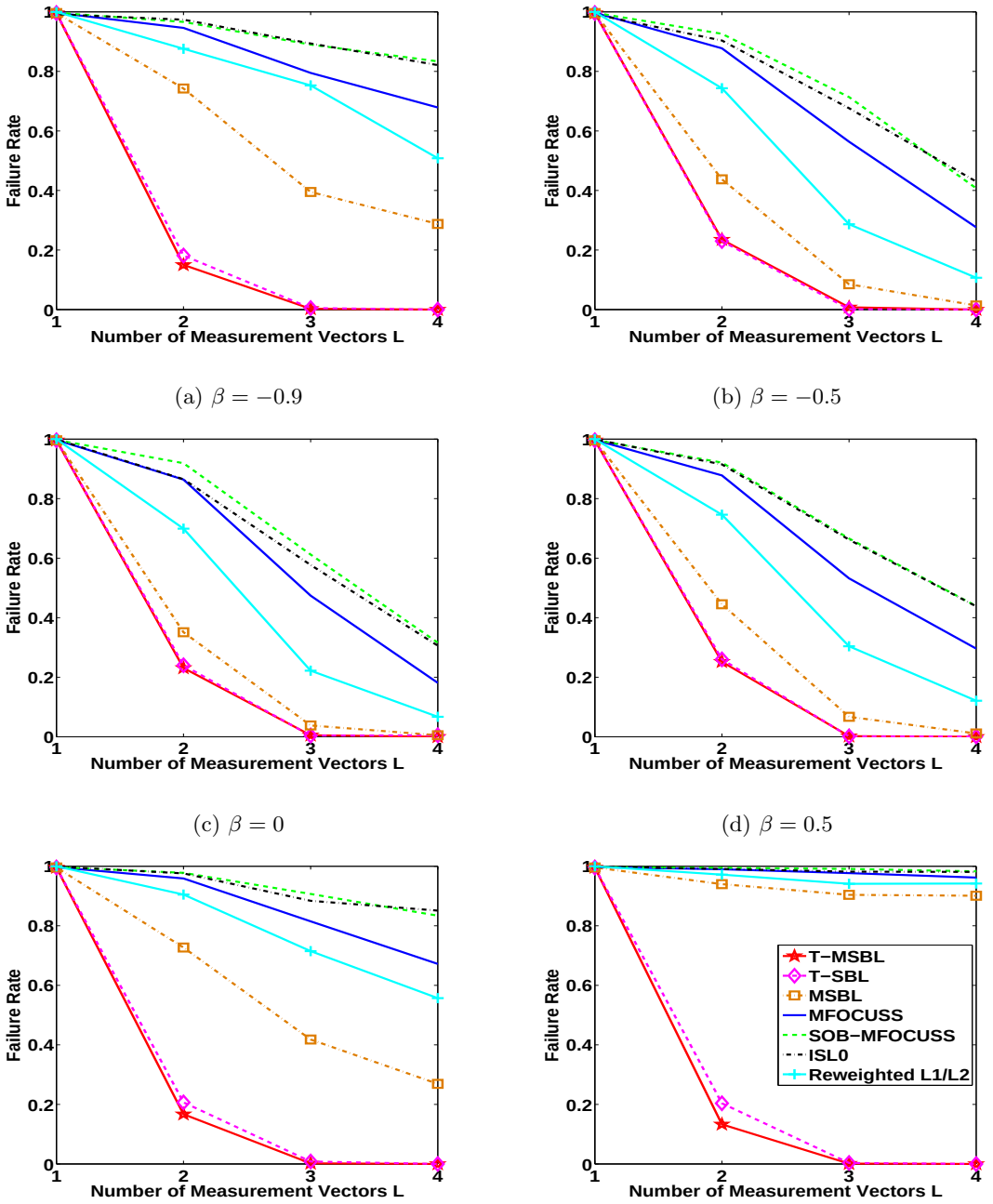
Figure III.3 shows that with L increasing, all the algorithms had better performance. But as $|\beta| \to 1$, for all the compared algorithms the benefit from multiple measurement vectors diminished. One surprising observation is that our T-MSBL and T-SBL had excellent performance in all cases, no matter what the temporal correlation was. Notice that even sources had no temporal correlation ($\beta = 0$), T-MSBL and T-SBL still had better performance than MSBL.

Next we compare all the algorithms in noisy environments. We set SNR = 25dB while kept other experimental settings unchanged. The behaviors of all the algorithms were similar to the noiseless case. To save space, we only present the cases of $\beta = 0.7$ and $\beta = 0.9$ in Figure III.4.

Since the performance of all the algorithms at a given correlation level β is the same as their performance at the correlation level $-\beta$, in the following we mainly show their performance at positive correlation levels.

III.E.2 Recovered Source Number at Different Temporal Correlation Levels

In this simulation we study the effects of temporal correlation on the number of accurately recovered sources in a noiseless case. The dictionary matrix $\mathbf{\Phi}$ was of the size 25×125. L was 4. K varied from 10 to 18. The sources were generated in the same manner as before. Algorithms were compared at four different temporal correlation levels, i.e. $\beta = 0$, 0.5, 0.9, and 0.99. Results (Figure III.5) show that T-



(a) $\beta = -0.9$

(b) $\beta = -0.5$

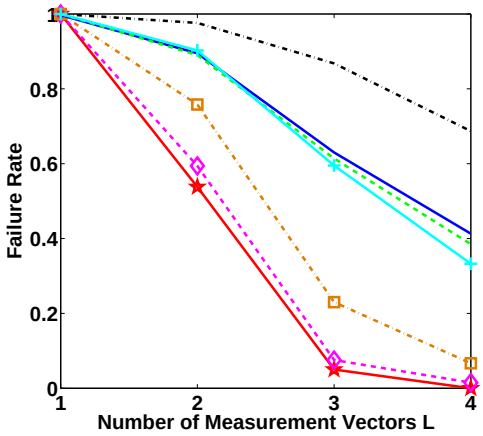
(c) $\beta = 0$

(d) $\beta = 0.5$

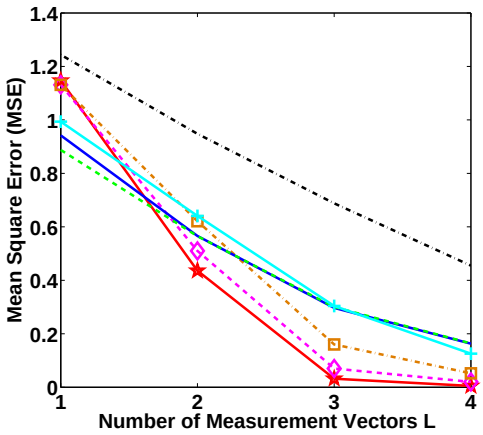
(e) $\beta = 0.9$

(f) $\beta = 0.99$

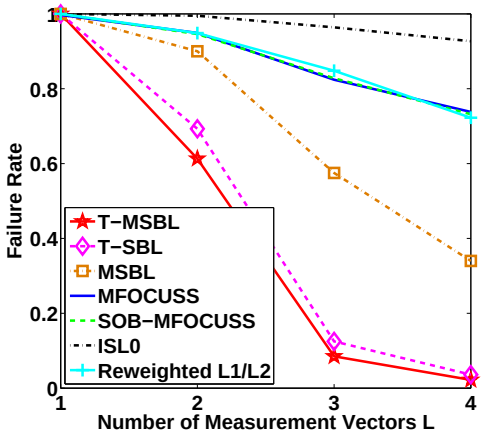
Figure III.3 Performance of all the algorithms at different temporal correlation levels when L varied from 1 to 4.



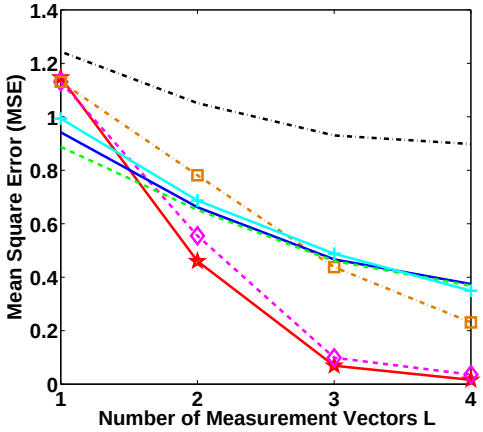
(a) $\beta = 0.7$



(b) $\beta = 0.7$

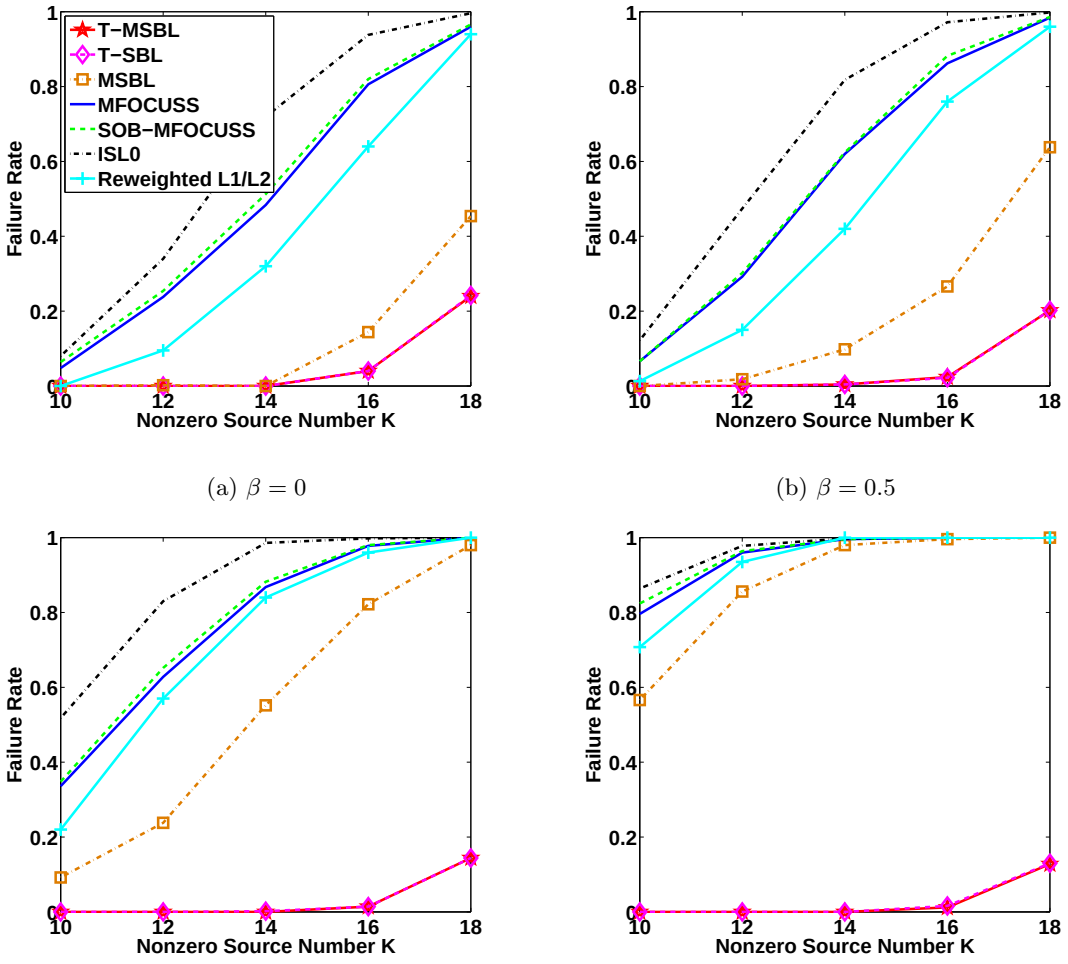


(c) $\beta = 0.9$



(d) $\beta = 0.9$

Figure III.4 The failure rate and the MSE of all the algorithms at different temporal correlation levels when L varied from 1 to 4 and SNR was 25 dB.



(a) $\beta = 0$

(b) $\beta = 0.5$

(c) $\beta = 0.9$

(d) $\beta = 0.99$

Figure III.5 Failure rates of all the algorithms when K varied from 10 to 18 at different temporal correlation levels.

MSBL and T-SBL accurately recovered much more sources than other algorithms, especially at high temporal correlation levels. This indicates that our proposed algorithms are very advantageous in the cases when the source number is large.

III.E.3 Ability to Handle Highly Underdetermined Problem

Most published works only compared algorithms in mildly underdetermined cases, namely, the ratio of N/M was about $2 \sim 5$. However, in some applications

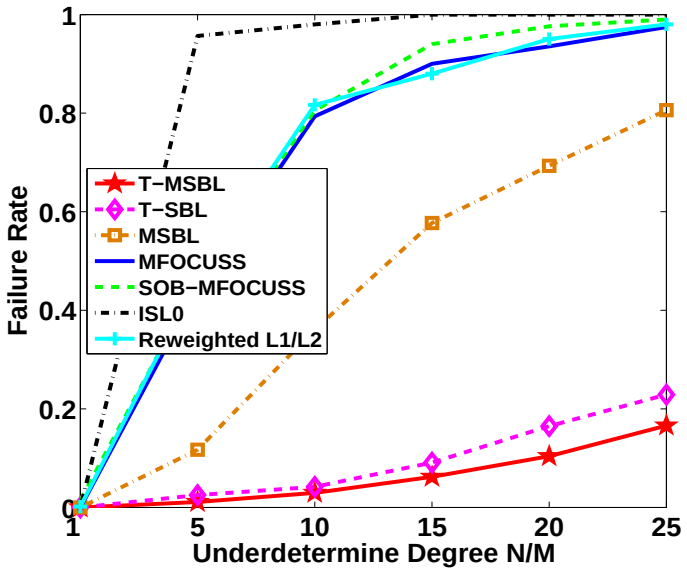


Figure III.6 Performance comparison in highly underdetermined cases when SNR was 25 dB.

such as neuroimaging, one can easily have $M \approx 100$ and $N \approx 100000$. Thus, in this simulation we compared the algorithms in the highly underdetermined cases when M was fixed at 25 and N/M varied from 1 to 25. The source number K was 12, and the measurement vector number L was 4. SNR was 25 dB. Different to previous simulations, all the sources were AR(1) processes but with different AR coefficients. Their AR coefficients were uniformly chosen from $(0.5, 1)$ at random. Results are presented in Figure III.6, from which we can see that when $N/M \geq 10$, all the compared algorithms had large errors. In contrast, our proposed algorithms had much lower errors. Note that due to the performance trade-off between M and N, if one increases M, algorithms can keep the same recovery performance for larger N/M.

III.E.4 Recovery Performance for Different Kinds of Sources

In previous simulations all the sources were AR(1) processes. Although we have pointed out that for small L modeling sources by AR(1) processes is

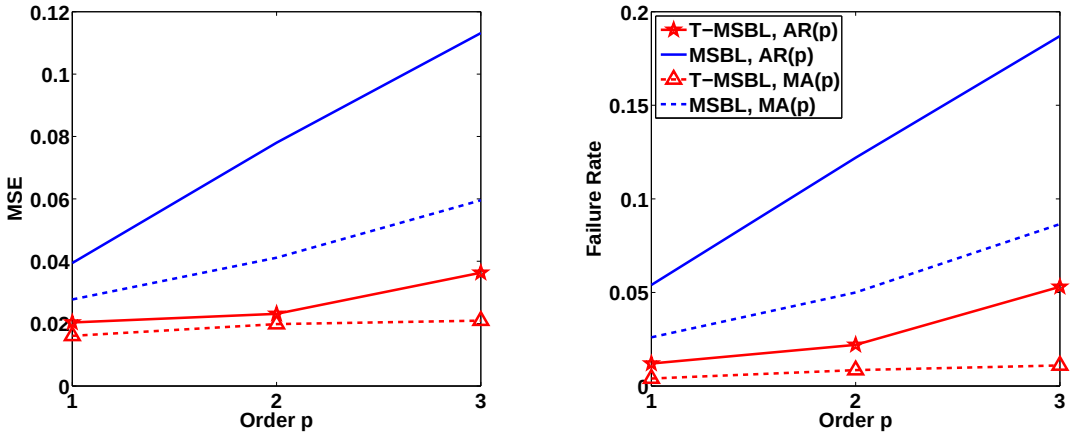
sufficient, here we carry out an experiment to show our algorithms maintaining the same superiority for various time series. Since from previous experiments we have seen that T-SBL has similar performance to T-MSBL, and that MSBL has the best performance among the compared algorithms, in this experiment we only compare T-MSBL with MSBL.

The dictionary matrix was of the size 25×125. L was 4. K was 14. SNR was 25dB. First we generated sources as three kinds of AR processes, i.e. AR(p) ($p = 1, 2, 3$). All the AR coefficients were randomly uniformly chosen from the feasible regions such that the processes were stable. We examined the algorithms' performance as a function of the AR order p. Results are given in Figure III.7, showing that T-MSBL again outperformed MSBL. With large p, the performance gap between the two algorithms increased. We repeated the previous experiment with the same experiment settings except that we replaced the AR(p) sources by moving-averaging sources MA(p) ($p = 1, 2, 3$). The MA coefficients were uniformly chosen from $(0, 1]$ at random. Again, we obtained the same results. These results imply that our algorithms maintain their superiority for various temporally structured sources, not only AR processes.

III.E.5 Recovery Ability at Different Noise Levels

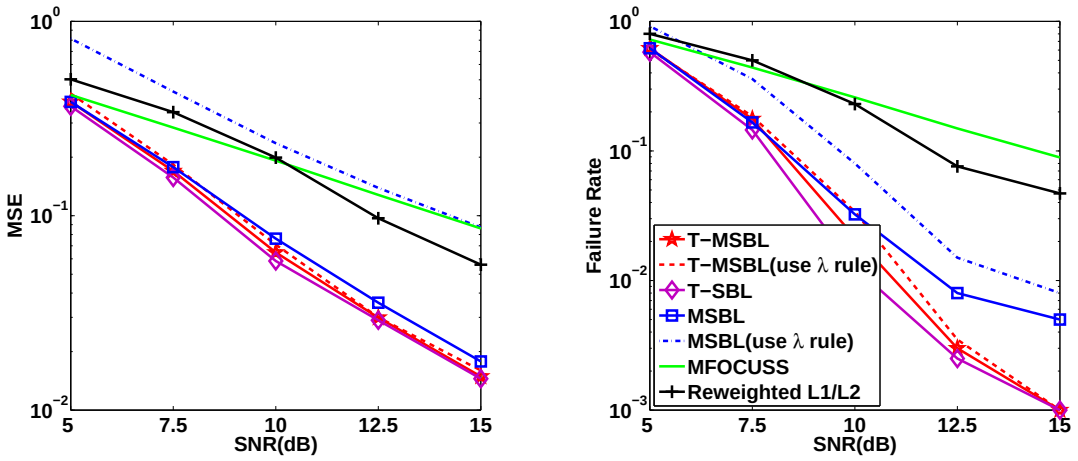
From previous experiments we have seen that the proposed algorithms significantly outperformed all the compared algorithms in noiseless scenarios and mildly noisy cases, even though to derive T-MSBL we used the approximation (III.27) which takes the equal sign only when $\mathbf{B} = \mathbf{I}$ (no temporal correlation) or $\lambda = 0$ (no noise). Some natural questions may be raised: What is the performance of T-SBL and T-MSBL in strongly noisy cases? Is it still beneficial to exploit temporal correlation in these cases? To answer these questions, we carry out the following experiment.

The dictionary matrix was of the size 25×125. The number of measurement vectors L was 4. The source number K was 7. All the sources were AR(1) processes



(a) (b)

Figure III.7 Performance of T-MSBL and MSBL for different AR(p) sources and different MA(p) sources measured in terms of MSE and failure rates.



(a) (b)

Figure III.8 Performance comparison at different noise levels. (a) shows the results in terms of MSE. (b) shows the results in terms of failure rates.

and the temporal correlation of each source was 0.8. SNR varied from 5 dB to 15 dB. The experiment was repeated 2000 trials. We compared the proposed T-SBL, T-MSBL with three representative algorithms, i.e. MSBL, MFOCUSS, and Reweighted ℓ_1/ℓ_2.

Note that in low SNR cases, the estimated $\mathbf{B}$ of T-SBL and T-MSBL can include large errors, and thus the estimated amplitudes of sources are distorted. To reduce the distortion, we set $\mathbf{B} = \mathbf{I}$ once the number of nonzero γ_i was less than N during the learning procedure. The reason is that the role of $\mathbf{B}$ is to prevent T-SBL/T-MSBL from arriving at local minima; once the algorithms approach global minima very closely, $\mathbf{B}$ is no longer useful.

Also note that the λ learning rules of T-SBL, T-MSBL and MSBL may not lead to optimal performance in low SNR cases. To avoid the potential disturbance of these λ learning rules, we provided the three SBL algorithms with the optimal λ^*'s, which were obtained by the exhaustive search method stated previously.

Figure III.8 shows that T-SBL and T-MSBL outperformed other algorithms in all the noise levels. This implies that even in low SNR cases exploiting temporal correlation of sources is beneficial.

But we want to emphasize that although the λ learning rules of the three SBL algorithms may not be optimal in low SNR cases, our proposed λ learning rules can lead to near-optimal performance, compared to the one of MSBL. To see this, we ran T-MSBL and MSBL again, but this time both algorithms used their λ learning rules. T-MSBL used the modified version of the λ learning rule (III.42), i.e. setting the off-diagonal elements of $\mathbf{\Phi\Gamma\Phi}^T$ to zeros. The results (Figure III.8) show that MSBL had very poor performance when using its λ learning rule. In contrast, T-MSBL's performance was very close to its performance when using its optimal λ^* [17]. The results indicate our proposed algorithms are advantageous in practical applications, since in practice the optimal λ^*'s are difficult to obtain, if not impossible.

III.E.6 Temporal Correlation: Beneficial or Detrimental?

From previous experiments one may think that temporal correlation is always harmful to algorithms' performance, at least not helpful. However, in this

[17] T-SBL had the same behavior. But for clarity we do not present its performance curve.

experiment we will show that when SNR is high, the performance of our proposed algorithms increases with increasing temporal correlation.

We set $M = 25$, $L = 4$, $K = 14$, and SNR $= 50$dB. The underdeterminacy ratio N/M varied from 5 to 20. Sources were generated as AR(1) processes with the common AR coefficient β. We considered the performance of T-MSBL and MSBL in three cases, i.e. the temporal correlation β was 0, 0.5, and 0.9, respectively. Results are shown in Figure III.9. As expected, the performance of MSBL deteriorated with increasing temporal correlation. But the behavior of T-MSBL was rather counterintuitive. It is surprising that the best performance of T-MSBL was not achieved at $\beta = 0$, but at $\beta = 0.9$. Clearly, high temporal correlation enabled T-MSBL to handle more highly underdetermined problems. For example, its performance at $N/M = 20$ with $\beta = 0.9$ was better than that at $N/M = 15$ with $\beta = 0.5$ or $\beta = 0$. The same phenomenon was observed in noiseless cases as well, and was observed for T-SBL.

The results indicating that temporal correlation is helpful may appear counterintuitive at first glance. A closer examination of the sparse recovery problems indicates a plausible explanation. There are two elements to the sparse recovery task; one is the location of the nonzero entries and the other is the value for the nonzero entries. Both tasks interact and combine to determine the overall performance. Correlation helps the estimation of the values for the nonzero entries and this may be important for the problem when dealing with finite matrices and may be lost when dealing with limiting results as the matrix dimension go to infinity. A more rigorous study of the interplay between estimation of the values and estimation of the locations is an interesting topic.

III.E.7 An Extreme Experiment on the Importance of Exploiting Temporal Correlation

It may be natural to take for granted that in noiseless cases, when source vectors are almost identical, algorithms have almost the same performance as in

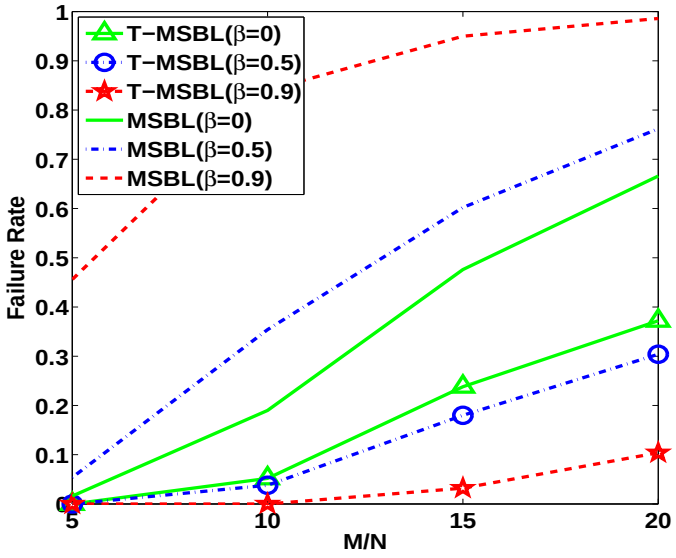
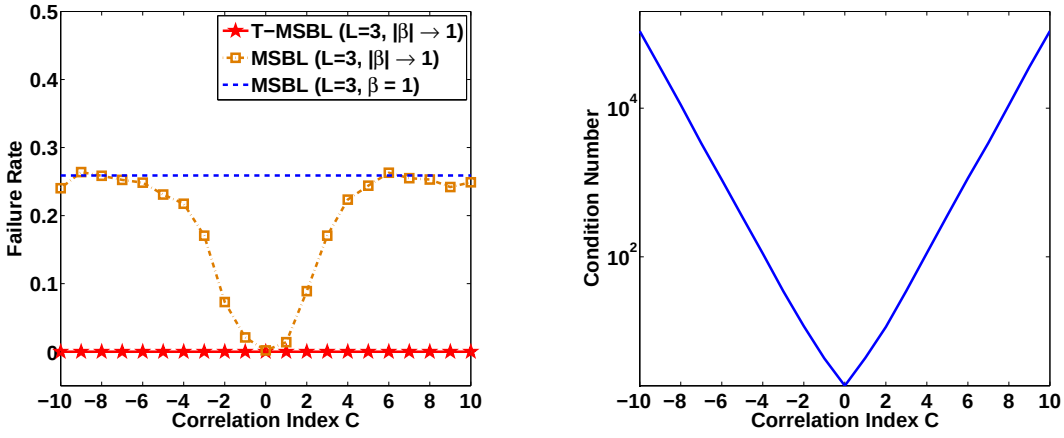


Figure III.9 Behaviors of MSBL and T-MSBL at different temporal correlation levels when SNR = 50dB.



(a) (b)

Figure III.10 (a) The performance and (b) the condition numbers of the submatrix formed by sources when the temporal correlation approximated to 1. The temporal correlation $\beta = \text{sign(C)}(1-10^{-|\text{C}|})$, where C was the correlation index varying from -10 to 10.

the case when only one measurement vector is available. In the following we show that it is not the case.

We designed a noiseless experiment. First, we generated a Hadamard matrix of the size 128×128. From the matrix, 40 rows were randomly selected in each trial and formed a dictionary matrix of the size 40×128. The source number K was 12, and the measurement vector number L was 3. Sources were generated as AR(1) processes with the common AR coefficient β, where $\beta = \text{sign}(C)(1 - 10^{-|C|})$. We varied C from -10 to 10 in order to see how algorithms behaved when the absolute temporal correlation, $|\beta|$, approximated to 1.

Figure III.10 (a) shows the performance curves of T-MSBL and MSBL when $|\beta| \to 1$, and also shows the performance curve of MSBL when $\beta = 1$. We observe an interesting phenomenon. First, as $|\beta| \to 1$, MSBL's performance closely approximated to its performance in the case of $\beta = 1$. It seems to make sense, because when $|\beta| \to 1$, every source vector provides almost the same information on locations and amplitudes of nonzero elements. Counter-intuitively, no matter how close $|\beta|$ was to 1, the performance of T-MSBL did not change. Figure III.10 (b) shows the averaged condition numbers of the submatrix formed by the sources (i.e. nonzero rows in $\mathbf{X}_{\text{gen}}$) at different correlation levels. We can see that the condition numbers increased with the increasing temporal correlation. This suggests that T-MSBL was not sensitive to the ill-condition issue in the source matrix, while MSBL is very sensitive. Although not shown here, we found that T-SBL had the same behavior as T-MSBL, while other MMV algorithms had the same behaviors as MSBL. The phenomenon was also observed when using other dictionary matrices, such as random Gaussian matrices.

These results emphasize the importance of exploiting the temporal correlation, and also motivate future theoretical studies on the temporal correlation and the ill-condition issue of source matrices.

III.F Discussions

Although there are a few works trying to exploit temporal correlation in the MMV model, based on our knowledge no works have explicitly studied the effects of temporal correlation, and no existing algorithms are effective in the presence of such correlation. Our work is a starting point in the direction of considering temporal correlation in the MMV model. However, there are many issues that are unclear so far. In this section we discuss some of them.

III.F.1 The Matrix B: Trade-off Between Accurately Modeling and Preventing Overfitting

In our algorithm development we used one single matrix $\mathbf{B}$ as the covariance matrix (up to a scalar) for each source model in order to avoid overfitting. Mathematically, it is straightforward to extend our algorithms to use multiple matrices to capture the covariance structures of sources. For example, one can classify sources into several groups, say G groups, and the sources in a group are all assigned by a common matrix $\mathbf{B}_i$ $(i = 1, \cdots, G,\ G \ll M)$ as the covariance matrix (up to a scalar). It seems that this extension can better capture the covariance structures of sources while still avoiding overfitting. However, we find that this extension (even for $G = 2$) has much poorer performance than our proposed algorithms and MSBL. One possible reason is that during the early stage of the learning procedure of our algorithms, the estimated sources from each iteration are far from the true sources, and thus grouping them based on their covariance structures is difficult, if not impossible. The grouping error may cause avalanche effect, leading to the noted poor performance. Reducing the grouping error and more accurately capturing the temporal correlation structures is an area for future work.

However, as we have stated, $\mathbf{B}$ plays a role of whitening each source. In our recent work [173, 170] we found that the operation $\mathbf{X}_{i\cdot}\mathbf{B}^{-1}\mathbf{X}_{i\cdot}^T$ $(\forall i)$ can replace the row-norms (such as the ℓ_2 norm and the ℓ_∞ norm) in iterative reweighted

ℓ_2 and ℓ_1 algorithms for the MMV model, functioning as a row regularization. This indicates that using one single matrix $\mathbf{B}$ may be a better method than using multiple matrices $\mathbf{B}_1, \cdots, \mathbf{B}_G$.

On the other hand, there may be many ways to parameterize and estimate $\mathbf{B}$. In this work we provide a general method to estimate $\mathbf{B}$. In [172] we proposed a method to parameterize $\mathbf{B}$ by a hyperparameter β, i.e.,

$$\mathbf{B} = \begin{bmatrix} 1 & \beta & \cdots & \beta^{L-1} \\ \beta & 1 & \cdots & \beta^{L-2} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{L-1} & \beta^{L-2} & \cdots & 1 \end{bmatrix}$$

which equivalently assumes the sources are AR(1) processes with the common AR coefficient β. The resulting algorithms have good performance as well. Also, for low SNR cases in our experiments, we added an identity matrix (with a scalar) to the estimated $\mathbf{B}$ in T-MSBL, and achieved satisfying performance. All these imply that $\mathbf{B}$ could have many forms. Finding the forms that are advantageous in strongly noisy environments is an important issue and needs further study.

III.F.2 The Parameter λ: Noise Variance or Regularization Parameter?

In our algorithms the covariance matrix of the multi-channel noise $\mathbf{V}_{\cdot i}$ ($i = 1, \cdots, L$) is $\lambda \mathbf{I}_M$ with the implicit assumption that each channel noise has the same variance λ. It is straightforward to extend our algorithms to consider the general noise covariance matrix $\mathrm{diag}([\lambda_1, \cdots, \lambda_M])$, i.e. assuming different channel noise have different variance. However, this largely increases parameters to estimate, and thus we may once again encounter an overfitting problem (similar to the overfitting problem in learning the matrix $\mathbf{B}_i$).

Some works [152, 105] considered alternative noise covariance models. In [105] the authors assumed that the covariance matrix of multi-channel noise is $\lambda \mathbf{C}$, instead of $\lambda \mathbf{I}_N$, where $\mathbf{C}$ is a known positive definite and symmetric matrix and λ

is an unknown noise-variance parameter. This model may better capture the noise covariance structures, but generally one does not know the exact value of $\mathbf{C}$. Thus there is no clear benefit from this covariance model. In [152], instead of deriving a learning rule for the noise covariance inside the SBL framework, the authors estimated the noise covariance by a method independent of the SBL framework. But this method is based on a large number of measurement vectors, and has a high computational load.

On the other hand, due to the works in [155, 149], which connected SBL algorithms to traditional convex relaxation methods such as Lasso [129] and Basis Pursuit Denoising [21], it was found that λ is functionally the same as the regularization parameters of those convex relaxation algorithms. This suggests the use of methods such as the modified L-curve procedure [108] or the cross-validation [129, 21] to choose λ especially in strongly noisy environments. It is also interesting to see that SBL algorithms could adopt the continuation strategies [11, 58], used in Lasso-type algorithms, to adjust the value of λ for better recovery performance or faster speed.

However, if some channels contain very large noise (e.g. outliers) and the number of such channels is very small, then as suggested in [156], we can extend the dictionary matrix $\mathbf{\Phi}$ to $[\mathbf{\Phi}, \mathbf{I}]$ and perform any sparse signal recovery algorithms without modification. The estimated 'sources' associated with the identity dictionary matrix are these large noise components.

III.F.3 Connections to Other Models

The time-varying sparsity model [140, 179] is another related model. Different to our MMV model that assumes the support of each source vector is the same, the time-varying sparsity model assumes the support is slowly time-varying. It is interesting to note that this model can be approximated by concatenation of several MMV models, where in each MMV model the support does not change. Thus our proposed T-SBL and T-MSBL can be used for this model. The results

are appealing, as shown in our recent work [170].

It should be noted that the proposed algorithms can be directly used for the SMV model. In this case the matrix $\mathbf{B}$ reduces to a scalar, and the γ_i learning rules are the same as the one in the basic SBL algorithm [153]. But due to the effective λ learning rules, our algorithms are superior to the basic SBL algorithm, especially in noisy cases.

III.G Conclusion

In this chapter, we addressed a multiple measurement vector (MMV) model in practical scenarios, where the source vectors are temporally correlated and the number of measurement vectors is small due to the common sparsity constraint. We showed that existing algorithms have poor performance when temporal correlation is present, and thus they have limited ability in practice. To solve this problem, we adopted the block sparse Bayesian learning framework, proposed in the previous chapter, which allows for easily modeling the temporal correlation and incorporating this information into derived algorithms. Based on this framework, we derived three algorithms, namely, T-SBL, T-MSBL, and T-MSBL-FP. The latter two can be seen as an extension of MSBL by replacing the ℓ_2 norm imposed on each source with a Mahalanobis distance measure. We connected the proposed algorithms to existing iterative reweighted ℓ_1 algorithms and iterative reweighted ℓ_2 algorithms. Motivated by this connection, we suggested a strategy to improve these iterative reweighted algorithms such that they can exploit the temporal correlation for better performance. Extensive experiments have shown that the proposed algorithms have superior performance to many state-of-the-art algorithms. Theoretical analysis also has shown that the proposed algorithms have desirable global and local minimum properties.

III.H Acknowledgements

The text of Chapter III, in full, is based on the material as it appears in: Zhilin Zhang and Bhaskar D. Rao, "Sparse Signal Recovery with Temporally Correlated Source Vectors Using Sparse Bayesian Learning", IEEE Journal of Selected Topics in Signal Processing, 2011, Zhilin Zhang and Bhaskar D. Rao, "Iterative Reweighted Algorithms for Sparse Signal Recovery with Temporally Correlated Source Vectors", in Proc. of the 36th International Conference on Acoustics, Speech, and Signal Processing, 2011, and Jing Wan, Zhilin Zhang, Jingwen Yan, Taiyong Li, Bhaskar D. Rao, Shiaofen Fang, Sungeun Kim, Shannon Risacher, Andrew Saykin, Li Shen, "Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer's Disease", in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2012. The dissertation author was a primary researcher and author of the cited papers.

III.I Appendix

III.I.1 Outline of the Proof of Theorem 1

Since the proof is a generalization of the Theorem 1 in [152], we only give an outline.

For convenience we consider the equivalent model (III.2). Let $\widehat{\mathbf{x}}$ be computed using $\widehat{\mathbf{x}} = (\lambda\widehat{\mathbf{\Sigma}}_0^{-1} + \mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{y}$ with $\widehat{\mathbf{\Sigma}}_0 = \text{diag}\{\widehat{\gamma}_1\widehat{\mathbf{B}}_1, \cdots, \widehat{\gamma}_N\widehat{\mathbf{B}}_N\}$, and $\widehat{\boldsymbol{\gamma}} \triangleq [\widehat{\gamma}_1, \cdots, \widehat{\gamma}_N]$ is obtained by globally minimizing the cost function for given $\widehat{\mathbf{B}}_i\,(\forall i)$ [18]:

$$\mathcal{L}(\gamma_i) = \mathbf{y}^T\mathbf{\Sigma}_y^{-1}\mathbf{y} + \log|\mathbf{\Sigma}_y|.$$

It can be shown [152] that when $\lambda \to 0$ (noiseless case), the above problem is

[18] In the proof we fix $\widehat{\mathbf{B}}_i$ because we will see $\widehat{\mathbf{B}}_i$ has no effect on the global minimum property.

equivalent to

$$\min : \qquad g(\mathbf{x}) \triangleq \min_{\boldsymbol{\gamma}} \left[\mathbf{x}^T \boldsymbol{\Sigma}_0^{-1} \mathbf{x} + \log |\boldsymbol{\Sigma}_y|\right] \tag{III.83}$$

$$\text{s.t.} : \qquad \mathbf{y} = \mathbf{D}\mathbf{x} \tag{III.84}$$

So we only need to show the global minimizer of (III.83) satisfies the property stated in the theorem.

Assume in the noiseless problem $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X}$, $\boldsymbol{\Phi}$ satisfies the URP condition [57]. For its any solution $\widehat{\mathbf{X}}$, denote the number of nonzero rows by K. Thus following the method in [152], we can show the above $g(\mathbf{x})$ satisfies

$$g(\mathbf{x}) = \mathcal{O}(1) + \big(ML - \min[ML, KL]\big) \log \lambda, \tag{III.85}$$

providing $\widehat{\mathbf{B}}_i$ is full rank. Here we adopt the notation $f(s) = \mathcal{O}(1)$ to indicate that $|f(s)| < C_1$ for all $s < C_2$, with C_1 and C_2 constants independent of s. Therefore, by globally minimizing (III.85), i.e. globally minimizing (III.83), K will achieve its minimum value, which will be shown to be K_0, the number of nonzero rows in $\mathbf{X}_{\text{gen}}$.

According to the result in [25, 36], if $\mathbf{X}_{\text{gen}}$ satisfies

$$K_0 < \frac{M+L}{2}$$

then there is no other solution (with K nonzero rows) such that $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X}$ with $K < \frac{M+L}{2}$. So, $K \geq K_0$, i.e. the minimum value of K is K_0. Once K achieves its minimum, we have $\widehat{\mathbf{X}} = \mathbf{X}_{\text{gen}}$.

In summary, the global minimum solution $\widehat{\boldsymbol{\gamma}}$ leads to the solution that equals to the unique sparsest solution $\mathbf{X}_{\text{gen}}$. And we can see, providing $\widehat{\mathbf{B}}_i$ is full rank, it does not affect the conclusion.

III.I.2 Proof of Lemma 2

Re-write the equation $\mathbf{y}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{y} = C$ by $\mathbf{y}^T \mathbf{u} = C$, where $\mathbf{u} \triangleq \boldsymbol{\Sigma}_y^{-1} \mathbf{y} = \big(\lambda \mathbf{I} + \mathbf{D}\boldsymbol{\Sigma}_0 \mathbf{D}^T\big)^{-1} \mathbf{y}$, from which we have $\mathbf{y} - \lambda \mathbf{u} = \mathbf{D}\boldsymbol{\Sigma}_0 \mathbf{D}^T \mathbf{u} = \mathbf{D}(\boldsymbol{\Gamma} \otimes \mathbf{B})\mathbf{D}^T \mathbf{u} = \mathbf{D}(\mathbf{I}_N \otimes$

$\mathbf{B})(\mathbf{\Gamma}\otimes\mathbf{I}_L)\mathbf{D}^T\mathbf{u} = \mathbf{D}(\mathbf{I}_N\otimes\mathbf{B})\text{diag}(\mathbf{D}^T\mathbf{u})\text{diag}(\mathbf{\Gamma}\otimes\mathbf{I}_L) = (\mathbf{\Phi}\otimes\mathbf{B})\text{diag}(\mathbf{D}^T\mathbf{u})(\boldsymbol{\gamma}\otimes\mathbf{1}_L)$. It can be seen that the matrix $\mathbf{A} \triangleq (\mathbf{\Phi}\otimes\mathbf{B})\text{diag}(\mathbf{D}^T\mathbf{u})$ is full row rank.

III.I.3 Proof of Theorem 2

The proof follows along the lines of Theorem 2 in [153] using our Lemma 1 and Lemma 2. Consider the optimization problem:

$$\begin{cases} \min: & f(\boldsymbol{\gamma}) \triangleq \log|\lambda\mathbf{I} + \mathbf{D}\mathbf{\Sigma}_0\mathbf{D}^T| \\ \text{s.t.}: & \mathbf{A}\cdot(\boldsymbol{\gamma}\otimes\mathbf{1}_L) = \mathbf{b} \\ & \boldsymbol{\gamma} \succeq \mathbf{0} \end{cases} \tag{III.86}$$

where $\mathbf{A}$ and $\mathbf{b}$ are defined in Lemma 2. From Lemma 1 and Lemma 2 we can see the optimization problem (III.86) is optimizing a concave function over a closed, bounded convex polytope. Obviously, any local minimum of $\mathcal{L}$, e.g. $\boldsymbol{\gamma}^*$, must also be a local minimum of the above optimization problem with $C = \mathbf{y}^T\big(\lambda\mathbf{I} + \mathbf{D}(\mathbf{\Gamma}^*\otimes\mathbf{B})\mathbf{D}^T\big)^{-1}\mathbf{y}$, where $\mathbf{\Gamma}^* \triangleq \text{diag}(\boldsymbol{\gamma}^*)$. Based on the Theorem 6.5.3 in [81] the minimum of (III.86) is achieved at an extreme point. Further, based on the Theorem in Chapter 2.5 of [81] the extreme point is a BFS to

$$\begin{cases} \mathbf{A}\cdot(\boldsymbol{\gamma}\otimes\mathbf{1}_L) = \mathbf{b} \\ \boldsymbol{\gamma} \succeq \mathbf{0} \end{cases}$$

which indicates $\|\boldsymbol{\gamma}\|_0 \leq ML$.

III.I.4 Proof of Lemma 3

For convenience we first consider the case of $K = M$. Let $\widetilde{\boldsymbol{\gamma}}$ be the vector consisting of nonzero elements in $\widehat{\boldsymbol{\gamma}}$, and $\widetilde{\mathbf{\Phi}}$ be a matrix consisting of the columns of $\mathbf{\Phi}$ whose indexes are the same as those of nonzero elements in $\widehat{\boldsymbol{\gamma}}$. Thus, the equation $\mathbf{Y} = \mathbf{\Phi}\widehat{\mathbf{X}}$ can be rewritten as $\mathbf{Y} = \widetilde{\mathbf{\Phi}}\widetilde{\mathbf{X}}$. By transferring it to its equivalent block sparse Bayesian learning model, we have $\mathbf{y} = \widetilde{\mathbf{D}}\widetilde{\mathbf{x}}$, where $\mathbf{y} \triangleq \text{vec}(\mathbf{Y}^T)$, $\widetilde{\mathbf{D}} \triangleq \widetilde{\mathbf{\Phi}}\otimes\mathbf{I}_L$, and $\widetilde{\mathbf{x}} \triangleq \text{vec}(\widetilde{\mathbf{X}}^T)$. Since $\widetilde{\mathbf{D}}$ is a square matrix with full rank, we have $\widetilde{\mathbf{x}} = \widetilde{\mathbf{D}}^{-1}\mathbf{y}$.

For convenience, let $\widetilde{\mathbf{x}}_i \triangleq \widetilde{\mathbf{x}}_{[(i-1)L+1:iL]}$, i.e. $\widetilde{\mathbf{x}}_i$ consists of elements of $\widetilde{\mathbf{x}}$ with indexes from $(i-1)L+1$ to iL. Now consider the cost function $\mathcal{L}$, which becomes

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}) &= \sum_{i=1}^{N}\Big(\frac{\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i}{\widetilde{\gamma}_i}+L\log\widetilde{\gamma}_i\Big)+M\log|\mathbf{B}| \\ &\quad +2\log|\widetilde{\mathbf{D}}|.\end{aligned}$$

Letting $\frac{\partial\mathcal{L}(\boldsymbol{\gamma})}{\partial\widetilde{\gamma}_i}=0$ gives

$$\widetilde{\gamma}_i=\frac{1}{L}\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i,\quad i=1,\cdots,K$$

The second derivative of $\mathcal{L}$ at $\widetilde{\gamma}_i=\frac{1}{L}\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i$ is given by

$$\frac{\partial^2\mathcal{L}(\boldsymbol{\gamma})}{\partial\widetilde{\gamma}_i^2}\Big|_{\widetilde{\gamma}_i=\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i}=\frac{\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i}{\widetilde{\gamma}_i^3}.$$

Since $\mathbf{B}$ is positive definite and $\widetilde{\mathbf{x}}_i\neq\mathbf{0}$, $\frac{\widetilde{\mathbf{x}}_i^T\mathbf{B}^{-1}\widetilde{\mathbf{x}}_i}{\widetilde{\gamma}_i^3}>0$. So $\widetilde{\gamma}_i=\frac{1}{L}\widetilde{\mathbf{x}}_i^T\widehat{\mathbf{B}}^{-1}\widetilde{\mathbf{x}}_i$ $(i=1,\cdots,K)$ is a local minimum.

If $\|\widehat{\boldsymbol{\gamma}}\|_0\triangleq K<M$, which implies there exists $\widetilde{\mathbf{x}}\in\mathbb{R}^{KL\times1}$ such that $\mathbf{y}=\widetilde{\mathbf{D}}\widetilde{\mathbf{x}}$, then we can expand the matrix $\widetilde{\mathbf{D}}$ to a full-rank square matrix $[\widetilde{\mathbf{D}},\mathbf{D}_e]$ by adding an arbitrary full column-rank matrix $\mathbf{D}_e$. And we expand $\widetilde{\mathbf{x}}$ to $[\widetilde{\mathbf{x}}^T,\boldsymbol{\varepsilon}^T]^T$, where $\boldsymbol{\varepsilon}\in\mathbb{R}^{(M-K)L\times1}$ and $\boldsymbol{\varepsilon}\to\mathbf{0}$. Therefore, $[\widetilde{\mathbf{D}},\mathbf{D}_e][\widetilde{\mathbf{x}}^T,\boldsymbol{\varepsilon}^T]^T\to\widetilde{\mathbf{D}}\widetilde{\mathbf{x}}=\mathbf{y}$. Similarly, we also expand $\widetilde{\boldsymbol{\gamma}}$ to $[\widetilde{\boldsymbol{\gamma}}^T,\boldsymbol{\zeta}^T]^T$ with $\boldsymbol{\zeta}\to\mathbf{0}$. Then, following the above steps, we can obtain the same result. Therefore, we finish the proof.

Chapter IV

Sparse Bayesian Learning Exploiting Spatio-Temporal Correlation

In Chapter I we have introduced the spatiotemporal sparse model:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} + \mathbf{V}, \tag{IV.1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times L}$, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M < N)$, and $\mathbf{X} \in \mathbb{R}^{N\times L}$. Further, the matrix $\mathbf{X}$ is assumed to have the following structure:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{[1]\cdot} \\ \mathbf{X}_{[2]\cdot} \\ \vdots \\ \mathbf{X}_{[g]\cdot} \end{bmatrix} \tag{IV.2}$$

where $\mathbf{X}_{[i]\cdot} \in \mathbb{R}^{d_i\times L}$ is the i-th block of $\mathbf{X}$, and $\sum_{i=1}^{g} d_i = N$. $\{d_1, \cdots, d_g\}$ is the block partition. Among the g blocks, only a few are nonzero blocks. The key assumption is that each block $\mathbf{X}_{[i]\cdot}(\forall i)$ is assumed to have spatiotemporal correlation. In other words, entries in each column of $\mathbf{X}_{[i]\cdot}$ are correlated (intra-block correlation), and entries in each row of $\mathbf{X}_{[i]\cdot}$ are also correlated (temporal correlation).

In the following we derive several SBL algorithms for the spatiotemporal sparse model.

Some specific notations are needed to pay attention to. For a matrix $\mathbf{A}$, $\mathbf{A}_{i\cdot}$ denotes the i-th row, and $\mathbf{A}_{\cdot j}$ denotes the j-th column. $\mathbf{A}_{[i]j}$ denotes the i-th block in the j-th column. $\mathbf{A}_{i[j]}$ denotes the j-th block in the i-th row. $\mathbf{A}_{[i]\cdot}$ denotes the i-th block of all the columns, while $\mathbf{A}_{\cdot[j]}$ denotes the j-th block of all the rows. $\mathbf{A}_{[k]}$ denotes the k-th diagonal block in $\mathbf{A}$.

IV.A Spatiotemporal SBL Model

Now we describe the spatiotemporal sparse model from a Bayesian perspective. To facilitate the algorithm development, we make the same assumptions as in the standard multivariate Bayesian variable selection model [14] (or called the conjugate multivariate linear regression model [32]). The i-th block $\mathbf{X}_{[i]\cdot}$ is

assumed to have the parameterized Gaussian distribution $p(\text{vec}(\mathbf{X}_{[i]\cdot}^T); \gamma_i, \mathbf{B}, \mathbf{A}_i) = \mathcal{N}(\mathbf{0}, (\gamma_i \mathbf{A}_i) \otimes \mathbf{B})$. Here $\mathbf{B} \in \mathbb{R}^{L \times L}$ is an unknown positive definite matrix capturing the correlation structure in each row of $\mathbf{X}_{[i]\cdot}$. The matrix $\mathbf{A}_i \in \mathbb{R}^{d_i \times d_i}$ is an unknown positive definite matrix capturing the correlation structure in each column of $\mathbf{X}_{[i]\cdot}$. γ_i is an unknown nonnegative scalar determining whether the i-th block is a zero block or not. Assuming the blocks $\{\mathbf{X}_{[i]\cdot}\}_{i=1}^g$ are mutually independent, the distribution of the matrix $\mathbf{X}$ can be expressed as

$$p(\text{vec}(\mathbf{X}^T); \mathbf{B}, \{\gamma_i, \mathbf{A}_i\}_i) = \mathcal{N}(\mathbf{0}, \mathbf{\Pi} \otimes \mathbf{B}) \tag{IV.3}$$

with

$$\mathbf{\Pi} \triangleq \begin{bmatrix} \gamma_1 \mathbf{A}_1 & & & \\ & \gamma_2 \mathbf{A}_2 & & \\ & & \ddots & \\ & & & \gamma_g \mathbf{A}_g \end{bmatrix}. \tag{IV.4}$$

Further, each row of the noise matrix $\mathbf{V}$ has the distribution $p(\mathbf{V}_{i\cdot}; \lambda, \mathbf{B}) = \mathcal{N}(\mathbf{0}, \lambda \mathbf{B})$, where λ is an unknown scalar. Assuming the rows are mutually independent, the distribution of $\mathbf{V}$ can be expressed as

$$p(\text{vec}(\mathbf{V}^T); \lambda, \mathbf{B}) = \mathcal{N}(\mathbf{0}, \lambda \mathbf{I} \otimes \mathbf{B}). \tag{IV.5}$$

Remark 1: The model is a combination of our previous works on the MMV model exploiting temporal correlation [174] and on the SMV model exploiting spatial intra-block correlation [171]. Thus the proposed model (IV.1) with the associated assumptions and the probability modelings is called *the spatiotemporal sparse Bayesian learning (STSBL) model*. The following section will present an algorithm which alternatively operates in the temporal domain and in the spatial domain.

Remark 2: The block partition $\{d_1, d_2, \cdots, d_g\}$ in (IV.2) is determined by users. To recover non-sparse signals as in the applications of compressed sensing of physiological signals (see Chapter VII), the design of the block partition could

be rather arbitrary, while the recovery performance is almost not affected. This property has been shown in [167] on the BSBL framework. The reason is that in our models (both the BSBL model and the STSBL model), the block partition is a kind of regularization, which helps estimate the covariance matrix of each column of $\mathbf{X}$, which, as a result, helps improve the estimate of $\mathbf{X}$.

Remark 3: Note that $\mathbf{X}$ and $\mathbf{V}$ share the common matrix $\mathbf{B}$ for modeling the correlation structure of each row. This is a widely used setting in Bayesian variable selection models [14].

IV.B STSBL-EM: Spatiotemporal SBL Algorithm Based on the EM Method

Motivated by the MMV work in Chapter III, where the matrix $\mathbf{B}$ can be viewed as a temporal whitening matrix, reducing the negative effect caused by temporal correlation, we propose a *switching-learning approach*, where the parameters $\{\gamma_i, \mathbf{A}_i\}_{i=1}^{g}$ and λ are estimated from a temporally whitened model, and the parameter $\mathbf{B}$ is estimated from a spatially whitened model. The resulting algorithm switches the estimation between the two models until convergence.

IV.B.1 Learning in the Temporally Whitened Model

To facilitate algorithm development, we first assume $\mathbf{B}$ is known. Letting $\widetilde{\mathbf{Y}} \triangleq \mathbf{Y}\mathbf{B}^{-\frac{1}{2}}$, $\widetilde{\mathbf{X}} \triangleq \mathbf{X}\mathbf{B}^{-\frac{1}{2}}$, and $\widetilde{\mathbf{V}} \triangleq \mathbf{V}\mathbf{B}^{-\frac{1}{2}}$, the original STSBL model (IV.1) becomes

$$\widetilde{\mathbf{Y}} = \mathbf{\Phi}\widetilde{\mathbf{X}} + \widetilde{\mathbf{V}}, \tag{IV.6}$$

where the columns of $\widetilde{\mathbf{X}}$ are independent, and so does $\widetilde{\mathbf{V}}$. Thus, the algorithm development becomes easier.

First, we have the prior for $p(\widetilde{\mathbf{X}};\mathbf{\Pi})$ and $p(\widetilde{\mathbf{V}};\lambda)$:

$$p(\widetilde{\mathbf{X}};\mathbf{\Pi}) = \prod_{i=1}^{L} p(\widetilde{\mathbf{X}}_{\cdot i}) \sim \prod_{i} \mathcal{N}(\mathbf{0},\mathbf{\Pi}) \tag{IV.7}$$

$$p(\widetilde{\mathbf{V}};\lambda) = \prod_{i=1}^{L} p(\widetilde{\mathbf{V}}_{\cdot i}) \sim \prod_{i} \mathcal{N}(\mathbf{0},\lambda\mathbf{I}) \tag{IV.8}$$

Then we have the likelihood:

$$p(\widetilde{\mathbf{Y}}_{\cdot i}|\widetilde{\mathbf{X}}_{\cdot i};\lambda) = \mathcal{N}(\mathbf{\Phi}\widetilde{\mathbf{X}}_{\cdot i},\lambda\mathbf{I}) \quad \forall i \tag{IV.9}$$

Thus, we obtain the posterior:

$$p(\widetilde{\mathbf{X}}_{\cdot i}|\widetilde{\mathbf{Y}}_{\cdot i};\lambda,\mathbf{\Pi}) = \mathcal{N}(\boldsymbol{\mu}_{\cdot i},\mathbf{\Sigma}) \quad \forall i \tag{IV.10}$$

with the mean $\boldsymbol{\mu}_{\cdot i}$ and the covariance matrix $\mathbf{\Sigma}$ given by

$$\boldsymbol{\mu}_{\cdot i} = \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\widetilde{\mathbf{Y}}_{\cdot i} \quad \forall i \tag{IV.11}$$

$$\mathbf{\Sigma} = (\mathbf{\Pi}^{-1}+\frac{1}{\lambda}\mathbf{\Phi}^T\mathbf{\Phi})^{-1} \tag{IV.12}$$

$$= \mathbf{\Pi}-\mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{\Pi} \tag{IV.13}$$

Once the parameters $\mathbf{\Pi}$ and λ are estimated, the MAP estimate of $\widetilde{\mathbf{X}}$ is directly given by the mean of the posterior, i.e.,

$$\widetilde{\mathbf{X}} \leftarrow \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\widetilde{\mathbf{Y}}, \tag{IV.14}$$

and the solution matrix $\mathbf{X}$ in the original STSBL model (IV.1) can be obtained:

$$\mathbf{X} \leftarrow \widetilde{\mathbf{X}}\mathbf{B}^{\frac{1}{2}}. \tag{IV.15}$$

Thus, estimating the parameters $\mathbf{\Pi}$ and λ is crucial to the algorithm. We use the the expectation maximization (EM) method to estimate them. In the EM method, $\widetilde{\mathbf{X}}$ is treated as hidden variable. The Q-function for estimating $\{\gamma_i\}_i$ and

$\{\mathbf{A}_i\}_i$ is

$$\begin{aligned}
\mathcal{Q}(\mathbf{\Pi}) &\triangleq E_{\widetilde{\mathbf{X}}|\widetilde{\mathbf{Y}};\Theta^{(\text{old})}}\big[\log p(\widetilde{\mathbf{X}};\{\gamma_i\}_i,\{\mathbf{A}_i\}_i)\big] \\
&= -\frac{L}{2}\log|\mathbf{\Pi}| - \frac{1}{2}\sum_{i=1}^{L} E_{\widetilde{\mathbf{X}}|\widetilde{\mathbf{Y}};\Theta^{(\text{old})}}\big[\widetilde{\mathbf{X}}_{\cdot i}^T\mathbf{\Pi}^{-1}\widetilde{\mathbf{X}}_{\cdot i}\big] \\
&= -\frac{L}{2}\sum_{i=1}^{g}\log|\gamma_i\mathbf{A}_i| - \frac{1}{2}\sum_{l=1}^{L}\mathrm{Tr}\Big[\mathbf{\Pi}^{-1}\big(\mathbf{\Sigma}+\boldsymbol{\mu}_{\cdot l}\boldsymbol{\mu}_{\cdot l}^T\big)\Big] \\
&= -\frac{L}{2}\sum_{i=1}^{g} d_i\log\gamma_i - \frac{L}{2}\sum_{i=1}^{g}\log|\mathbf{A}_i| \\
&\quad -\frac{1}{2}\sum_{l=1}^{L}\sum_{j=1}^{g}\frac{1}{\gamma_j}\mathrm{Tr}\Big[\mathbf{A}_j^{-1}\big(\mathbf{\Sigma}_{[j]}+\boldsymbol{\mu}_{[j]l}\boldsymbol{\mu}_{[j]l}^T\big)\Big], \qquad \text{(IV.16)}
\end{aligned}$$

where $\Theta^{(\text{old})}$ denotes all the parameters estimated in the previous iteration, $\mathbf{\Sigma}_{[j]}$ denotes the j-th diagonal block in $\mathbf{\Sigma}$ which corresponds to the j-th block in $\widetilde{\mathbf{X}}$, and $\boldsymbol{\mu}_{[j]l}$ denotes the j-th block in the l-th column of $\boldsymbol{\mu}$.

Setting to zero the derivative of (IV.16) w.r.t. γ_i, we obtain the updating rule for γ_i:

$$\gamma_i \leftarrow \frac{1}{Ld_i}\sum_{l=1}^{L}\mathrm{Tr}\Big[\mathbf{A}_i^{-1}\big(\mathbf{\Sigma}_{[i]}+\boldsymbol{\mu}_{[i]l}\boldsymbol{\mu}_{[i]l}^T\big)\Big]. \qquad \text{(IV.17)}$$

Setting to zero the derivative of (IV.16) w.r.t. $\mathbf{A}_i$, we obtain the updating rule for $\mathbf{A}_i$:

$$\mathbf{A}_i \leftarrow \frac{1}{L}\sum_{l=1}^{L}\frac{\mathbf{\Sigma}_{[i]}+\boldsymbol{\mu}_{[i]l}\boldsymbol{\mu}_{[i]l}^T}{\gamma_i}. \qquad \text{(IV.18)}$$

Note that one can further regularize the estimate as shown later.

To estimate λ, the Q-function is given by

$$\begin{aligned}
\mathcal{Q}(\lambda) &= E_{\widetilde{\mathbf{X}}|\widetilde{\mathbf{Y}};\Theta^{(\text{old})}}\big[\log p(\widetilde{\mathbf{Y}}|\widetilde{\mathbf{X}};\lambda)\big] \\
&\propto -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}E_{\widetilde{\mathbf{X}}|\widetilde{\mathbf{Y}};\Theta^{(\text{old})}}\Big[\sum_{l=1}^{L}\|\widetilde{\mathbf{Y}}_{\cdot l} - \mathbf{\Phi}\widetilde{\mathbf{X}}_{\cdot l}\|_2^2\Big] \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\sum_{l=1}^{L}\Big[\|\widetilde{\mathbf{Y}}_{\cdot l} - \mathbf{\Phi}\boldsymbol{\mu}_{\cdot l}\|_2^2 \\
&\quad + E_{\widetilde{\mathbf{X}}|\widetilde{\mathbf{Y}};\Theta^{(\text{old})}}\big[\|\mathbf{\Phi}(\widetilde{\mathbf{X}}_{\cdot l} - \boldsymbol{\mu}_{\cdot l})\|_2^2\big]\Big] \\
&= -\frac{ML}{2}\log\lambda - \frac{1}{2\lambda}\|\widetilde{\mathbf{Y}} - \mathbf{\Phi}\boldsymbol{\mu}\|_{\mathcal{F}}^2 - \frac{L}{2\lambda}\text{Tr}\big(\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{\Phi}\big). \qquad \text{(IV.19)}
\end{aligned}$$

Setting its derivative to zero, we have

$$\lambda \leftarrow \frac{1}{ML}\|\widetilde{\mathbf{Y}} - \mathbf{\Phi}\boldsymbol{\mu}\|_{\mathcal{F}}^2 + \frac{1}{M}\text{Tr}\big(\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{\Phi}\big). \qquad \text{(IV.20)}$$

Similar as in [171], at low SNR cases the above updating rule should be modified to

$$\lambda \leftarrow \frac{1}{ML}\|\widetilde{\mathbf{Y}} - \mathbf{\Phi}\boldsymbol{\mu}\|_{\mathcal{F}}^2 + \frac{1}{M}\sum_{i=1}^{g}\text{Tr}\big(\mathbf{\Sigma}_{[i]}\mathbf{\Phi}_{\cdot[i]}^T\mathbf{\Phi}_{\cdot[i]}\big), \qquad \text{(IV.21)}$$

where $\mathbf{\Phi}_{\cdot[i]}$ denotes the consecutive columns in $\mathbf{\Phi}$ which correspond to the i-th block in $\widetilde{\mathbf{X}}$. In noiseless scenarios one can simply set $\lambda = 10^{-10}$ or other small values, instead of performing the above updating rules.

In the above development we have assumed that $\mathbf{B}$ is given. This parameter can be estimated in a spatially whitened model discussed below.

IV.B.2 Learning in the Spatially Whitened Model

To estimate the matrix $\mathbf{B}$, we consider the following equivalent form of the original model (IV.1):

$$\mathbf{Y} = \overline{\mathbf{\Phi}} \cdot \overline{\mathbf{X}} + \mathbf{V} \qquad \text{(IV.22)}$$

where $\overline{\mathbf{\Phi}} \triangleq \mathbf{\Phi}\mathbf{A}^{\frac{1}{2}}$ and $\overline{\mathbf{X}} \triangleq \mathbf{A}^{-\frac{1}{2}}\mathbf{X}$. In this model, $\overline{\mathbf{X}}$ maintains the same block structure as $\mathbf{X}$, but its each block has no intra-block correlation due to the spatially whitening effect from $\mathbf{A}^{-\frac{1}{2}}$. Thus, in this model estimating $\mathbf{B}$ becomes easier.

Following the approach to derive the T-MSBL algorithm [174] and assuming $\mathbf{X}$, $\{\gamma_i\}_i$ and $\{\mathbf{A}_i\}_i$ have been obtained from the temporally whitened model, we have the following updating rule for the matrix $\mathbf{B}$:

$$\begin{aligned}\check{\mathbf{B}} &\leftarrow \sum_{i=1}^{g}\gamma_i^{-1}\overline{\mathbf{X}}_{[i]\cdot}^T\overline{\mathbf{X}}_{[i]\cdot}+\lambda^{-1}(\mathbf{Y}-\overline{\mathbf{\Phi X}})^T(\mathbf{Y}-\overline{\mathbf{\Phi X}})\\ &= \sum_{i=1}^{g}\frac{\mathbf{X}_{[i]\cdot}^T\mathbf{A}_i^{-1}\mathbf{X}_{[i]\cdot}}{\gamma_i}+\frac{(\mathbf{Y}-\mathbf{\Phi X})^T(\mathbf{Y}-\mathbf{\Phi X})}{\lambda} \qquad \text{(IV.23)}\\ \mathbf{B} &\leftarrow \check{\mathbf{B}}/\|\check{\mathbf{B}}\|_{\mathcal{F}} \qquad \text{(IV.24)}\end{aligned}$$

where $\overline{\mathbf{X}}_{[i]\cdot}\in\mathbb{R}^{d_i\times L}$ is the i-th block in $\overline{\mathbf{X}}$, and $\overline{\mathbf{X}}_{[i]\cdot}\triangleq\mathbf{A}_i^{-\frac{1}{2}}\mathbf{X}_{[i]\cdot}$. Note that in noisy scenarios one must regularize the estimate as in [171]. However, for the task considered here the regularization is not needed.

We denote the above algorithm by STSBL-EM. Due to limited data, the estimates of $\mathbf{B}$ and $\{\mathbf{A}_i\}_i$ are needed to be regularized. This issue will be discussed in Chapter IV.D.

IV.C STSBL-BO: Spatiotemporal SBL Algorithm Based on the Bound-Optimization Method

In the previous section we have derived an EM based method. As we have known, the EM optimization method is slow. Thus, here we derive another algorithm based on the bound optimization method [42, 123]. We have used this optimization method to derive a block SBL algorithm in Chapter II.

We can use the switching-learning procedure to derive the algorithm. But here we choose to directly derive it from its cost function.

We first transform the original STSBL model to the following block sparse model

$$\mathbf{y}=\mathbf{D}\mathbf{x}+\mathbf{v} \qquad \text{(IV.25)}$$

where $\mathbf{y}=\text{vec}(\mathbf{Y}^T)\in\mathbb{R}^{ML\times 1}$, $\mathbf{D}=\mathbf{\Phi}\otimes\mathbf{I}_L$, $\mathbf{x}=\text{vec}(\mathbf{X}^T)\in\mathbb{R}^{NL\times 1}$, and $\mathbf{v}=\text{vec}(\mathbf{V}^T)$. Based on the probability models (IV.3) and (IV.5), we obtain

the posterior

$$p(\mathbf{x}|\mathbf{y};\Theta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{IV.26}$$

where Θ denotes all the parameters $\{\gamma_i, \mathbf{A}_i\}_i, \mathbf{B}, \lambda$. The covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\begin{aligned} \boldsymbol{\Sigma} &= \big((\boldsymbol{\Pi} \otimes \mathbf{B})^{-1} + \mathbf{D}^T(\lambda\mathbf{I} \otimes \mathbf{B})^{-1}\mathbf{D}\big)^{-1} \qquad \text{(IV.27)} \\ &= \big[\boldsymbol{\Pi} - \boldsymbol{\Pi}\boldsymbol{\Phi}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\Pi}\big] \otimes \mathbf{B} \end{aligned}$$

and the mean $\boldsymbol{\mu}$ is given by

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\Sigma}\mathbf{D}^T(\lambda\mathbf{I} \otimes \mathbf{B})^{-1}\mathbf{y} \qquad \text{(IV.28)} \\ &= (\boldsymbol{\Pi} \otimes \mathbf{B})\mathbf{D}^T\big(\lambda\mathbf{I} \otimes \mathbf{B} + \mathbf{D}(\boldsymbol{\Pi} \otimes \mathbf{B})\mathbf{D}^T\big)^{-1}\mathbf{y} \qquad \text{(IV.29)} \\ &= \mathrm{vec}\big(\mathbf{Y}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\Pi}\big) \end{aligned}$$

Therefore, once all the parameters Θ are estimated, the MAP estimate of $\mathbf{X}$ is given by the posterior mean:

$$\mathbf{X} = \boldsymbol{\Pi}\boldsymbol{\Phi}^T(\lambda\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T)^{-1}\mathbf{Y} \tag{IV.30}$$

To estimate these parameters Θ, we use the Type II maximum likelihood [132], which leads to the following cost function

$$\begin{aligned} \mathcal{L}(\Theta) &= -2\log\int p(\mathbf{y}|\mathbf{x};\lambda)p(\mathbf{x};\{\gamma_i, \mathbf{A}_i\}, \mathbf{B})dx \\ &= \mathbf{y}^T\boldsymbol{\Sigma}_\mathbf{y}^{-1}\mathbf{y} + \log|\boldsymbol{\Sigma}_\mathbf{y}| \qquad \text{(IV.31)} \end{aligned}$$

where $\boldsymbol{\Sigma}_\mathbf{y} = \lambda\mathbf{I} \otimes \mathbf{B} + \mathbf{D}(\boldsymbol{\Pi} \otimes \mathbf{B})\mathbf{D}^T$. Now we derive learning rules for each of these parameters.

IV.C.1 Learning rule for γ_i

Note that the first term in the cost function (IV.31) is convex with respect to $\boldsymbol{\gamma}$, and the second term in the cost function is concave with respect to $\boldsymbol{\gamma}$. Since

the goal is to minimize the cost function, we consider an upper-bound for the second term, and then minimize the upper-bound of the cost function.

An upper-bound for the second term is its supporting hyperplane. Let $\boldsymbol{\gamma}^*$ be a given point in the $\boldsymbol{\gamma}$-space. We have

$$\log|\boldsymbol{\Sigma}_{\mathbf{y}}| \;\leq\; \sum_{i=1}^{g} \mathrm{Tr}\big((\boldsymbol{\Sigma}_{\mathbf{y}}^*)^{-1}\mathbf{D}_{\cdot[i]}(\mathbf{A}_i\otimes\mathbf{B})\mathbf{D}_{\cdot[i]}^T\big)(\gamma_i-\gamma_i^*) + \log|\boldsymbol{\Sigma}_{\mathbf{y}}^*| \quad \text{(IV.32)}$$

where $\boldsymbol{\Sigma}_{\mathbf{y}}^* = \boldsymbol{\Sigma}_{\mathbf{y}}|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^*}$, and $\mathbf{D}_{\cdot[i]} = \boldsymbol{\Phi}_{\cdot[i]}\otimes\mathbf{I}_L$, and $\boldsymbol{\Phi}_{\cdot[i]}$ is the i-th block of $\boldsymbol{\Phi}$ corresponding to $\mathbf{X}_{[i]\cdot}$. Besides, notice:

$$\begin{aligned}
& \mathbf{y}^T(\lambda\mathbf{I}\otimes\mathbf{B}+\mathbf{D}(\boldsymbol{\Pi}\otimes\mathbf{B})\mathbf{D}^T)^{-1}\mathbf{y} \\
\overset{(*)}{=}\; & \lambda^{-1}\mathbf{y}^T\Big\{\mathbf{I}\otimes\mathbf{B}^{-1} \\
& -(\mathbf{I}\otimes\mathbf{B}^{-1})\mathbf{D}\big[\lambda^{-1}\mathbf{D}^T\big(\mathbf{I}\otimes\mathbf{B}^{-1}\big)\mathbf{D}+\boldsymbol{\Pi}^{-1}\otimes\mathbf{B}^{-1}\big]^{-1}\mathbf{D}^T(\mathbf{I}\otimes\mathbf{B}^{-1})\lambda^{-1}\Big\}\mathbf{y} \\
\overset{(**)}{=}\; & \lambda^{-1}\mathbf{y}^T(\mathbf{I}\otimes\mathbf{B}^{-1})(\mathbf{y}-\mathbf{D}\boldsymbol{\mu}) \\
=\; & (\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})+\boldsymbol{\mu}^T\mathbf{D}^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}\mathbf{y} \\
& -\boldsymbol{\mu}^T\mathbf{D}^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}\mathbf{D}\boldsymbol{\mu} \\
\overset{(***)}{=}\; & (\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})+\boldsymbol{\mu}^T\big(\boldsymbol{\Sigma}^{-1}-\mathbf{D}^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}\mathbf{D}\big)\boldsymbol{\mu} \\
\overset{(****)}{=}\; & (\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})+\boldsymbol{\mu}^T(\boldsymbol{\Pi}\otimes\mathbf{B})^{-1}\boldsymbol{\mu} \qquad \text{(IV.33)}
\end{aligned}$$

where (*) used the matrix inversion lemma, (**) used (IV.28) and (IV.27), (***) used (IV.28), and (****) used (IV.27).

Substituting (IV.32) and (IV.33) into the cost function (IV.31), we obtain the upper bound:

$$\begin{aligned}
\mathcal{G}(\{\gamma_i\}_i) \;=\; & \frac{1}{\lambda}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu}) \\
+\; & \sum_{i=1}^{g}\mathrm{Tr}\big((\boldsymbol{\Sigma}_{\mathbf{y}}^*)^{-1}\mathbf{D}_{\cdot[i]}(\mathbf{A}_i\otimes\mathbf{B})\mathbf{D}_{\cdot[i]}^T\big)(\gamma_i-\gamma_i^*) \\
+\; & \boldsymbol{\mu}^T(\boldsymbol{\Pi}\otimes\mathbf{B})^{-1}\boldsymbol{\mu}+\log|\boldsymbol{\Sigma}_{\mathbf{y}}^*|. \qquad \text{(IV.34)}
\end{aligned}$$

Taking the derivative of $\mathcal{G}(\{\gamma_i\}_i)$ with respect to γ_i, we finally obtain the following

learning rule

$$\gamma_i \leftarrow \sqrt{\frac{\boldsymbol{\mu}_{[i]}^T(\mathbf{A}_i^{-1}\otimes\mathbf{B}^{-1})\boldsymbol{\mu}_{[i]}}{\mathrm{Tr}\big((\boldsymbol{\Sigma}_{\mathbf{y}})^{-1}\mathbf{D}_{\cdot[i]}(\mathbf{A}_i\otimes\mathbf{B})\mathbf{D}_{\cdot[i]}^T\big)}} \tag{IV.35}$$

where $\boldsymbol{\mu}_{[i]} \in \mathbb{R}^{d_iL\times 1}$, and the quantity $\boldsymbol{\Sigma}_{\mathbf{y}}^*$ in (IV.34) is replaced with $\boldsymbol{\Sigma}_{\mathbf{y}}$, keeping in mind that $\boldsymbol{\Sigma}_{\mathbf{y}}$ is calculated using the estimated parameters in the previous iteration. Note that the rule (IV.35) can be rewritten as

$$\gamma_i \leftarrow \sqrt{\frac{L^{-1}\mathrm{Tr}\big(\mathbf{X}_{[i]\cdot}\mathbf{B}^{-1}\mathbf{X}_{[i]\cdot}^T\mathbf{A}_i^{-1}\big)}{\mathrm{Tr}\big((\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}_{\cdot[i]}\mathbf{A}_i\boldsymbol{\Phi}_{\cdot[i]}^T\big)}} \tag{IV.36}$$

where $\mathbf{X}$ is estimated by (IV.30).

IV.C.2 Learning Rule for B

This learning rule can be easily obtained by noting that

$$\log|\boldsymbol{\Sigma}_{\mathbf{y}}| = L\log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T| + M\log|\mathbf{B}|$$

and using the result in (IV.33). Thus, the cost function (IV.31) becomes

$$\begin{aligned}\mathcal{L}(\mathbf{B}) &= L\log|\lambda\mathbf{I}+\boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T| + M\log|\mathbf{B}| \\ &\quad +(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu}) + \boldsymbol{\mu}^T(\boldsymbol{\Pi}\otimes\mathbf{B})^{-1}\boldsymbol{\mu}. \end{aligned}\tag{IV.37}$$

Note that

$$\frac{\partial\boldsymbol{\mu}^T(\boldsymbol{\Pi}\otimes\mathbf{B})^{-1}\boldsymbol{\mu}}{\partial\mathbf{B}} = -\sum_{i=1}^{g}\gamma_i^{-1}\mathbf{B}^{-1}\mathbf{X}_{[i]\cdot}^T\mathbf{A}_i^{-1}\mathbf{X}_{[i]\cdot}\mathbf{B}^{-1}$$

and

$$\frac{\partial}{\partial\mathbf{B}}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu})^T(\lambda\mathbf{I}\otimes\mathbf{B})^{-1}(\mathbf{y}-\mathbf{D}\boldsymbol{\mu}) = -\lambda^{-1}\mathbf{B}^{-1}(\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X})^T(\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X})\mathbf{B}^{-1}.$$

Thus, we obtain the learning rule:

$$\check{\mathbf{B}} \leftarrow \sum_{i=1}^{g}\frac{\mathbf{X}_{[i]\cdot}^T\mathbf{A}_i^{-1}\mathbf{X}_{[i]\cdot}}{\gamma_i} + \frac{(\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X})^T(\mathbf{Y}-\boldsymbol{\Phi}\mathbf{X})}{\lambda} \tag{IV.38}$$

$$\mathbf{B} \leftarrow \check{\mathbf{B}}/\|\check{\mathbf{B}}\|_{\mathcal{F}} \tag{IV.39}$$

where the goal of (IV.39) is to avoid the ambiguity among $\mathbf{A}_i, \gamma_i$ and $\mathbf{B}$. In (IV.38) the first term is data-related, while the second term is noise-related. When the noise is very small, or does not exist (i.e., $\lambda \to 0$), the second term can be removed for robustness. Alternatively, the estimate $\check{\mathbf{B}}$ can be further regularized, which will be discussed in Section IV.D. Note that the learning rule is the same as the one in STSBL-EM.

IV.C.3 Learning Rule for $\mathbf{A}_i$

From the original cost function (IV.31) or the equivalent one (IV.37), one can derive a learning rule for $\mathbf{A}_i(\forall i)$ as long as the condition $L \geq \max\{d_1, \cdots, d_g\}$ holds. Or, one can derive a learning rule for general situations, but it takes large computational load due to the coupling with $\mathbf{B}$. Thus, we consider to estimate $\mathbf{A}_i$ in the temporally whitened model as we have done in the development of STSBL-EM.

Assume $\mathbf{B}$ has been estimated. Letting $\widetilde{\mathbf{Y}} \triangleq \mathbf{Y}\mathbf{B}^{-\frac{1}{2}}$, $\widetilde{\mathbf{X}} \triangleq \mathbf{X}\mathbf{B}^{-\frac{1}{2}}$, and $\widetilde{\mathbf{V}} \triangleq \mathbf{V}\mathbf{B}^{-\frac{1}{2}}$, the original model (IV.1) becomes

$$\widetilde{\mathbf{Y}} = \mathbf{\Phi}\widetilde{\mathbf{X}} + \widetilde{\mathbf{V}}, \tag{IV.40}$$

where the columns of $\widetilde{\mathbf{X}}$ are independent, and so does $\widetilde{\mathbf{V}}$. Now the model is a block sparse Bayesian learning model [171] with multiple measurement vectors.

Following the EM method in [171, 174], we can easily derive the learning rule for $\mathbf{A}_i(\forall i)$:

$$\mathbf{A}_i \leftarrow \frac{1}{L}\sum_{l=1}^{L} \frac{\widetilde{\mathbf{\Sigma}}_{[i]} + \widetilde{\boldsymbol{\mu}}_{[i]l}\widetilde{\boldsymbol{\mu}}_{[i]l}^T}{\gamma_i}, \tag{IV.41}$$

where

$$\begin{aligned} \widetilde{\mathbf{\Sigma}} &= \mathbf{\Pi} - \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{\Pi} \\ \widetilde{\boldsymbol{\mu}} &= \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\mathbf{Y}\mathbf{B}^{-\frac{1}{2}}. \end{aligned}$$

For better results, the estimated $\mathbf{A}_i(\forall i)$ can be further regularized, which will be discussed in Chapter IV.D.

IV.C.4 Learning rule for λ

From the equivalent model (IV.40) we can derive the learning rule for λ using the EM method, as we have done for STSBL-EM. The learning rule is the same as the one in STSBL-EM.

IV.D Regularization

In our underdetermined spatiotemporal model the number of unknown parameters is much larger than the number of available data. Thus, regularization to the learning of the matrices $\mathbf{B}$ and $\{\mathbf{A}_i\}_i$ is very important. Suitable regularization helps to overcome the learning difficulty resulting from the very limited data.

As in [174], we can regularize the $\check{\mathbf{B}}$ in (IV.23) by

$$\check{\mathbf{B}} \leftarrow \sum_{i=1}^{g} \gamma_i^{-1} \mathbf{X}_{[i]\cdot}^T \mathbf{A}_i^{-1} \mathbf{X}_{[i]\cdot} + \eta \mathbf{I} \tag{IV.42}$$

where η is a positive scalar. This regularization is shown empirically to increase robustness in noisy environments. In noiseless environments, this regularization is not needed.

To regularize the estimates of $\{\mathbf{A}_i\}_i$, we use the strategy in [171], i.e., modeling each column in $\mathbf{X}_{[i]\cdot}$ as an AR(1) process with the common AR coefficient r for all i. The strategy can be summarized as follows.

- Step 1: Obtain the AR coefficient r_i from each $\mathbf{A}_i$:

  $$r_i \leftarrow \frac{m_1^i}{m_0^i}, \quad \forall i$$

  where m_0^i is the average of entries in the main diagonal of $\mathbf{A}_i$ and m_1^i is the average of entries in the main sub-diagonal of $\mathbf{A}_i$. Note that due to some numerical problems, $\frac{m_1^i}{m_0^i}$ may be out of the feasible range $(-1, 1)$, and thus further constrain may be imposed; for example, $r_i \leftarrow \text{sign}(\frac{m_1^i}{m_0^i}) \min\{|\frac{m_1^i}{m_0^i}|, 0.99\}$;

- Step 2: Average:

  $$r \leftarrow \frac{1}{g} \sum_{i=1}^{g} r_i$$

- Step 3: Reconstruct the regularized $\mathbf{A}_i$:

$$\mathbf{A}_i \leftarrow \begin{bmatrix} 1 & r & \cdots & r^{d_i-1} \\ \vdots & \vdots & & \vdots \\ r^{d_i-1} & r^{d_i-2} & \cdots & 1 \end{bmatrix} \quad \forall i$$

This method can be viewed as a simplified version of the one used in [172], where a gradient-descent method was used to estimate the AR coefficient, which results in huge computational load.

Note that the parameter-averaging strategy has been widely used in artificial neural network and the machine learning communities to overcome overfitting.

Experiments showed they helped further improve the algorithm's performance. In fact, denoting the true sparse solution by $\mathbf{X}_{\text{gen}}$ and the number of nonzero rows in $\mathbf{X}_{\text{gen}}$ by K_0, we have the following theorem:

Theorem 1: *In the limit as $\lambda \to 0$, assuming $K_0 < (M+L)/2$, the global minimum of the cost function (IV.31) is unique and produces an estimate which is equal to $\mathbf{X}_{\text{gen}}$, irrespective of the estimates of $\mathbf{B}$ and $\{\mathbf{A}_i\}_i$.*

It can be proved by straightforwardly following the Theorem 1 in Chapter III. This theorem implies that in noiseless situations the regularization strategies to $\mathbf{A}_i$ and $\mathbf{B}$ do not affect the global minimum of our algorithm in the sense that the global minimum corresponds to the true sparse solution. Thus, regularization strategies only affect the probability of our algorithm to converge to its local minima.

IV.E Experiment

This section gives an experiment on recovery of a compressed audio signal. More experiments can be found in Chapter VII and Chapter IX.

The length of the audio signal, $\mathbf{x}$, was 81920 data points, which was partitioned into T segments $\{\mathbf{x}_i\}_{i=1}^T$ of length N. Each segment $\mathbf{x}_i$ was compressed

Table IV.1 Performance comparison in terms of NMSE and runtime at different segment length N. The number in a parenthesis is runtime (in seconds), while the number outside a parenthesis is NMSE (in dB).

	$N = 512$	$N = 1024$	$N = 2048$	$N = 4096$	$N = 8192$
STSBL-EM	**-22.6** (**30.2** s)	-	-	-	-
SL0	-15.0 (16.6 s)	-16.2 (42.1 s)	-17.5 (129.7 s)	-20.4 (625.0 s)	-21.0 (2725 s)
EM-BG-AMP	-14.3 (50.2 s)	-16.0 (114.8 s)	-16.8 (172.0 s)	-19.3 (426.9 s)	-20.2 (1108 s)
SPGL-1	-13.7 (97.3 s)	-13.8 (89.9 s)	-16.5 (199.3 s)	-19.0 (510.1 s)	-19.8 (1541 s)
BCS	-13.0 (52.0 s)	-14.7 (68.4 s)	-16.0 (144.5 s)	-18.9 (422.8 s)	-18.4 (2270 s)
OMP	-12.4 (5.8 s)	-14.5 (19.4 s)	-14.7 (109.0 s)	-18.0 (521.6 s)	-
SP	-12.5 (31.7 s)	-14.2 (85.3 s)	-15.2 (308.0 s)	-17.5 (1325 s)	-

into $N/2$ samples, denoted by $\mathbf{y}_i$. The sensing matrix $\mathbf{\Phi}$ was a Gaussian random matrix.

Rather than directly recovering $\mathbf{x}_i$ from $\mathbf{y}_i$ and $\mathbf{\Phi}$, we considered to use the DCT to help recover. Namely, we first recovered the DCT coefficients of each segment via

$$\mathbf{y}_i = \mathbf{\Phi}\mathbf{D}\boldsymbol{\theta}_i$$

where $\mathbf{D}$ was the orthonormal basis of the DCT such that $\mathbf{x}_i = \mathbf{D}\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i$ was the DCT coefficients. Then we recovered $\mathbf{x}_i$ according to $\mathbf{x}_i = \mathbf{D}\boldsymbol{\theta}_i$.

In this experiment, we set $N = 512, 1024, 2048, 4096, 8192$, which corresponded to $T = 160, 80, 40, 20, 10$ segments, respectively. And then we performed six state-of-the-art algorithms. They were SL0 [94], EM-BG-AMP [141], SPGL-1 [137], BCS [70], OMP [134], and Subspace Pursuit (SP) [27]. Their NMSE and total runtime at different N are summarized in Table Table IV.1.

From the Table, we can see if we want to obtain good quality, we need to increase the segment length N. However, the cost is that the recovery time is significantly increased. For example, to achieve the quality of NMSE = -21dB, SL0 needed to choose a large segment length, which was 8192, but the total recovery time was 2725 seconds!

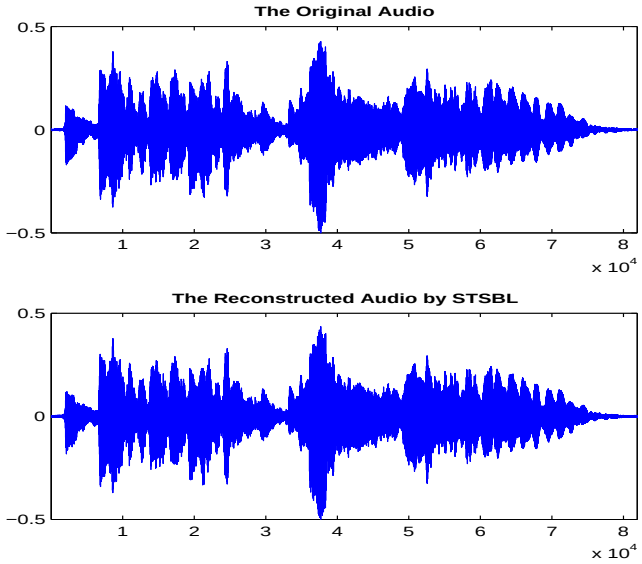


Figure IV.1 The waveforms of the original audio signal and of the recovered audio signal by STSBL-EM.

Next we performed the proposed STSBL-EM algorithm. We considered to jointly recover 8 segments of the length 512 at the same time (i.e., $L = 8$ and $N = 512$ in the spatiotemporal sparse model). The block size in the user-defined block partition was 16 (i.e., $d_i = 16(\forall i)$). The maximum number of iterations was set to 40. The resulting NMSE and total runtime are also summarized in Table IV.1. The recovered waveform and the original waveform are shown in Figure IV.1. Clearly, STSBL-EM had the higher recovery quality with NMSE = -22.6 dB, but only cost 30.2 seconds.

IV.F Conclusion

Spatiotemporal sparse model is a specific multiple measurement vector model, which can be viewed as a combination of the canonical block sparse model (see Chapter II) and the canonical multiple measurement vector model (see Chapter III). In this chapter, we proposed two spatiotemporal sparse Bayesian learning

algorithms for this model, which exploit the spatiotemporal correlation. The performance is confirmed by an experiment on recovery of a compressed audio signal. More experiments on real-world data can be found in Chapter VII and Chapter IX.

IV.G Acknowledgements

The text of Chapter IV, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Spatiotemporal Sparse Bayesian Learning with Applications to Compressed Sensing of Multichannel ECG for Wireless Telemonitoring", submitted for publication to IEEE Trans. on Biomedical Engineering, 2012, and Zhilin Zhang, Jing Wan, Shiaofen Fang, Andrew Saykin, Li Shen, "Correlation- and Nonlinearity-Aware Sparse Bayesian Learning with Applications to the Prediction of Cognitive Scores from Neuroimaging Measures", submitted to IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013. The dissertation author was a primary researcher and author of the cited papers.

Chapter V

Sparse Bayesian Learning for Signals with Time-Varying Sparsity

This chapter considers the time-varying sparse model, which is expressed as follows

$$\mathbf{y}(t) = \mathbf{\Phi}\mathbf{x}(t) + \mathbf{v}(t), \quad t = 1, 2, \cdots, L \tag{V.1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}(M \ll N)$ is a known matrix with columns assumed to satisfy certain conditions such as the Unique Representation Property (URP) condition [57]. Depending on applications, $\mathbf{\Phi}$ could be a random incoherent matrix (e.g. in MRI compression) or a deterministic coherent matrix (e.g. source localization). $\mathbf{y}(t) \in \mathbb{R}^{M \times 1}$ is the available measurement vector at time point t. $\mathbf{x}(t)$ is the unknown solution vector at time t. The number of nonzero entries in $\mathbf{x}(t)$ $(\forall t)$ has to be less than a threshold to ensure a unique global solution [36]. $\mathbf{v}(t)$ is the unknown noise vector at time t. In source localization, $\mathbf{y}(t)$ is the received signals by array sensors at time t, and $\mathbf{x}(t)$ is the source vector whose nonzero entries indicate active source signals at associated locations (or directions). In the model (V.1) a key characteristics is that the support of $\mathbf{x}(t)$ (i.e. indexes of nonzero entries in $\mathbf{x}(t)$) changes along time t.

If the support of $\mathbf{x}(t)$ does not change all the time, then the model (V.1) becomes the MMV model [1]. This may be the favorite case, since it has been proved that the failure probability of support recovery of $\mathbf{x}(t)(t = 1, \cdots, L)$ decreases exponentially with L, and many effective algorithms have been proposed for this model. However, when the support of $\mathbf{x}(t_i)$ is totally different from $\mathbf{x}(t_j)(\forall i \neq j)$, the model can be only treated as L separate basic SMV models. This may be the worst case, since in this case we cannot benefit from jointly exploiting multiple measurement vectors. Fortunately, in most applications the support of $\mathbf{x}(t)$ changes slowly. Such property can be exploited for better performance than treating the model (V.1) as L separate SMV models.

[1] Recall that in an MMV model, the matrix $\mathbf{X} \triangleq [\mathbf{x}(1), \cdots, \mathbf{x}(L)]$ is called the *source matrix*, and each row of $\mathbf{X}$ is called a *source*.

V.A Literature Review

A number of algorithms have been proposed [161, 114, 138, 139, 109] exploiting the property of slowly changing support. For example, in [138] a so-called LS-CS algorithm was proposed, which applies an SMV algorithm on the least squares residual computed using the estimate of the support from the previous time. A similar method was later proposed in [139], which uses the support estimate from the previous time and then finds the source vector at current time which satisfies the data constraint and is sparsest outside of that support. These algorithms all adopt an SMV algorithm to estimate current source vector. This may not ensure the estimation quality. Also, they heavily rely on estimates in the previous time. Once large errors occur in the previous time, these errors can propagate to the future estimation.

On the other hand, some people implicitly or explicitly exploited the MMV model by using the fact that several successive source vectors may have the common sparsity pattern [170, 163, 5]. In addition to the common sparsity pattern (see Chapter III), other properties can also be used, such as the amplitude smoothness of successive source vectors [163, 5]. Unfortunately, these algorithms blindly divide the whole data stream into a number of segments, each segment being treated as an MMV model. Also, they use deterministic ways to exploit the amplitude smoothness, such as defining a deterministic smoothness matrix or evaluating the total variation. These deterministic ways are not data-adaptive, and are empirically proved to be poor [174].

In the following we derive an online algorithm, called **Slide-TMSBL**, which exploits the common sparsity profile of successive source vectors. But different to existing algorithms, the algorithm also exploits temporal correlation of sources. In addition, it can automatically divide the data stream into segments such that each segment satisfies an MMV model, functioning like a change-point detection algorithm operating in the $\mathbf{x}$-space.

V.B The Slide-TMSBL Algorithm

In [170] we viewed the time-varying model (V.1) as a concatenation of several equal-length segments; each segment is treated as an MMV model. Then we performed T-MSBL on each segment. For convenience, we denote this method by **Segment-TMSBL**. Segment-SBL has been shown to have better performance than some state-of-the-art algorithms, such as LS-CS [138]. However, its main drawback is that the segment partition is blindly decided, which is inconsistent with the true partition. The simulation in [170] showed that different partitions lead to different performance.

To overcome this drawback, we propose the following algorithm. The basic idea is to slide a varying-length time-window over the measurement vector stream step by step along time. A decision-making mechanism ensures the source vectors in the time-window have the same support. The ending of the time-window is denoted by t_0, while its beginning is current time $t\,(t \geq t_0)$. The procedure is described as follows. After the time-window is updated (thus the source vectors in the time-window have the same support), T-MSBL is performed on measurement vectors in the time-window to obtain robust estimates of the source vectors, which fully benefits from the MMV model. Next, the time-window extends its beginning from t to $t+1$. Then the decision-making mechanism decides whether the support of the coming source vector $\mathbf{x}(t+1)$ changes or not. If not, the ending of the time-window, t_0, does not change; otherwise, t_0 is set to current time, i.e. $t_0 \leftarrow t+1$. Thereafter, the time-window is updated, and the procedure continues to the next time.

The basic flow of the algorithm is summarized in Algorithm 1. Some remarks are given in order:

(1) As shown in Chapter III, T-MSBL has a robust learning rule for λ. Generally, its estimate, $\widehat{\lambda}$, is three to five times larger than the true noise variance. So we set $\eta = \widehat{\lambda}/4$ such that η is close to the true noise variance. The denominator

Algorithm 1 Basic Flow

Input: measurement vectors $\mathbf{y}(1), \mathbf{y}(2), \cdots$; dictionary matrix $\mathbf{\Phi}$; initial time-window length $L_{\text{ini}} \geq 1$

Output: estimates of source vectors $\widehat{\mathbf{x}}(1), \widehat{\mathbf{x}}(2), \cdots$

Initialization: Apply T-MSBL on the measurement vectors in the initial time-window $\mathbf{y}(1), \cdots, \mathbf{y}(t_{L_{\text{ini}}})$, obtaining the estimates of sources vectors $\widehat{\mathbf{x}}(1), \cdots, \widehat{\mathbf{x}}(t_{L_{\text{ini}}})$ and saving them. Set $t_0 = 1$.

for $t = t_{L_{\text{ini}}} + 1, t_{L_{\text{ini}}} + 2, \cdots$ **do**

(1) Apply T-MSBL on the segment $\mathbf{y}(t_0), \cdots, \mathbf{y}(t)$, obtaining the results $\widehat{\mathbf{x}}(t_0), \cdots, \widehat{\mathbf{x}}(t)$ and the estimate of λ, $\widehat{\lambda}$.

(2) Calculate the residual at time t: $\mathbf{r} = \mathbf{y}(t) - \mathbf{\Phi}\widehat{\mathbf{x}}(t)$ and its variance $\text{var}(\mathbf{r})$.

(3) Set the threshold $\eta = \widehat{\lambda}/4$.

if $\text{var}(\mathbf{r}) < \eta$ **then**

(4) Save $\widehat{\mathbf{x}}(t_0), \cdots, \widehat{\mathbf{x}}(t)$.

else

(5) Apply T-MSBL on $\mathbf{y}(t)$, obtaining the estimate $\widehat{\mathbf{x}}(t)$ and save it.

(6) Set $t_0 = t$

end if

end for

4 is not crucial. Other values from 3 to 5 lead to similar performance.

(2) To find whether the support of the coming source vector $\mathbf{x}(t+1)$ changes or not, a natural strategy is to compare the support of $\widehat{\mathbf{x}}(t+1)$ to that of $\widehat{\mathbf{x}}(t)$. However, this strategy is not robust. Due to noise disturbance, even if the supports of $\mathbf{x}(t+1)$ and $\mathbf{x}(t)$ are the same, the estimates $\widehat{\mathbf{x}}(t+1)$ and $\widehat{\mathbf{x}}(t)$ can have slightly different supports. So we use the strategy presented in Step (1)-(2), which is based on the following observation. If $\mathbf{x}(t+1)$ has the same support as $\mathbf{x}(t_0), \cdots, \mathbf{x}(t)$, then the joint estimation of $\mathbf{x}(t_0), \cdots, \mathbf{x}(t+1)$ can achieve higher accuracy, since the failure probability of recover decreases exponentially with the increasing number of measurement vectors. Thus the variance $\text{var}(\mathbf{r})$ should be still below the threshold η. However, if $\mathbf{x}(t+1)$ has some nonzero elements whose locations outside of the support of $\mathbf{x}(t_0), \cdots, \mathbf{x}(t)$, then the joint estimation of $\mathbf{x}(t_0), \cdots, \mathbf{x}(t+1)$ results in poorer estimate of $\mathbf{x}(t+1)$ (and other source vectors). This is because the

locations of the extra nonzero elements in $\mathbf{x}(t+1)$ equivalently increase the nonzero rows in the solution matrix $[\mathbf{x}(t_0), \cdots, \mathbf{x}(t+1)]$, which T-MSBL tries to jointly recover. Increased nonzero rows in the solution matrix makes the recovery task more difficult [25]. Consequently, the recovery quality of $\mathbf{x}(t+1)$ deteriorates, and thus the variance $\text{var}(\mathbf{r})$ exceeds the threshold η.

(3) Although in Step (1) T-MSBL is performed on the segment from $\mathbf{y}(t_0)$ to $\mathbf{y}(t)$, the computational load is much close to the case when performed on only $\mathbf{y}(t)$. This is because T-MSBL can adopt the same strategies (SVD decomposition and the equivalent transformation of measurement vectors) as in [154] to reduce computational load.

(4) Note that the estimated value of λ is always updated at each time. One advantage of this is that the algorithm can be also used to the case when noise variance is time-varying. Based on our knowledge, no existing algorithms consider this case. However, if we know that the noise variance does not change, we can fix it to some value, such as an estimated value from a suitable time-window.

However, the algorithm has a flaw. When the algorithm finds the support of $\mathbf{x}(t)$ changes, it will perform T-MSBL only on $\mathbf{y}(t)$ (Step (5)). This corresponds to an SMV model, and the estimate of $\mathbf{x}(t)$ may have large errors. When a new measurement vector $\mathbf{y}(t+1)$ is available, the joint estimation of $\mathbf{x}(t)$ and $\mathbf{x}(t+1)$ may also have large errors (since in this case the corresponding MMV model only contains two measurement vectors). As a result, it probably determines $\mathbf{x}(t+1)$ has different support to $\mathbf{x}(t)$. Therefore, it again uses T-MSBL on only $\mathbf{y}(t+1)$ to estimate $\mathbf{x}(t+1)$, which, again, results in large errors in $\widehat{\mathbf{x}}(t+1)$. This vicious circle may continue for a long time. Note that this issue widely exists in the algorithms using SMV algorithms as their core, such as LS-CS [138] and modified-CS [139].

To solve this issue, we make some modifications. We use indicators, denoted by $\text{chanegPt}(t)$, to record change points of the supports of estimated source vector series. If the time t is a change point, set $\text{changePt}(t) = 1$; otherwise, set $\text{changePt}(t) = 0$. When $h_0 (h_0 \geq 1)$ *successive* change points are detected, the

algorithm applies T-MSBL on the measurement vectors corresponding to these change points. The estimated source vectors are stored as final estimates (replacing the previous estimates). The reason that T-MSBL is used here is that several successive change points indicate the estimates at these change points contain large errors (remind that the true support of source vectors changes slowly). The large errors are due to the use of SMV models at each change point. Thus, T-MSBL is used to jointly estimate these source vectors at these change points. This corresponds to an MMV model with h_0 measurement vectors, and the estimation error of each source vector is reduced largely. However, h_0 should not be too large. This is because in this case successive h_0 change points may not occur. Also, an MMV model with too many measurement vectors is at the risk of containing many nonzero rows in the solution matrix, which instead reduces the estimation quality. In our experiments we set $h_0 = 4$.

The algorithm, called **Slide-TMSBL**, is given in Algorithm 2. Advantages of Slide-TMSBL are summarized as follows.

(1) It largely exploits the advantages of the MMV model. By exploring the local stationarity of the supports of source vector series, it uses the MMV model to gain better recovery performance. This is one of the reasons that the Slide-TMSBL outperforms other algorithms which do not exploit the MMV model, such as LS-CS [138] and Modified-CS [139].

(2) Slide-TMSBL can automatically detect the change points of supports, dividing the data stream into segments such that each segment satisfies an MMV model. This ability ensures that it can better benefit from advantages of the MMV model.

(3) By applied T-MSBL to each segment, Slide-TMSBL can exploit temporal correlation of sources. In contrast, SOB-MFOCUSS [163] and some Lasso variants in [5] need users to design some penalty functions, which are not data-adaptive. It is found that adaptively learning temporal correlation can achieve better performance (see Chapter III). This is another reason that Slide-TMSBL

Algorithm 2 Slide-TMSBL

Input: measurement vectors $\mathbf{y}(1), \mathbf{y}(2), \cdots$; dictionary matrix $\mathbf{\Phi}$; initial time-window length $L_{\text{ini}} \geq 1$; user-defined parameter $h_0 \geq 1$

Output: estimates of source vectors $\widehat{\mathbf{x}}(1), \widehat{\mathbf{x}}(2), \cdots$; change-point indicators $\text{changePt}(1), \text{changePt}(2), \cdots$

Initialization: Apply T-MSBL on the measurement vectors in the initial time-window $\mathbf{y}(1), \cdots, \mathbf{y}(t_{L_{\text{ini}}})$, obtaining the estimates of sources vectors $\widehat{\mathbf{x}}(1), \cdots, \widehat{\mathbf{x}}(t_{L_{\text{ini}}})$ and saving them. Set $t_0 = 1$. Set $\text{changePt}(1), \cdots, \text{changePt}(t_{L_{\text{ini}}})$ to zeros.

for $t = t_{L_{\text{ini}}} + 1, t_{L_{\text{ini}}} + 2, \cdots$ **do**

(1) Apply T-MSBL on the segment $\mathbf{y}(t_0), \cdots, \mathbf{y}(t)$, obtaining the results $\widehat{\mathbf{x}}(t_0), \cdots, \widehat{\mathbf{x}}(t)$ and the estimate $\widehat{\lambda}$.

(2) Calculate the residual at time t: $\mathbf{r} = \mathbf{y}(t) - \mathbf{\Phi}\widehat{\mathbf{x}}(t)$ and its variance $\text{var}(\mathbf{r})$.

(3) Set the threshold $\eta = \widehat{\lambda}/4$.

if $\text{var}(\mathbf{r}) < \eta$ **then**

(4) Save $\widehat{\mathbf{x}}(t_0), \cdots, \widehat{\mathbf{x}}(t)$.

(5) Set $\text{changePt}(t_0 + 1), \cdots, \text{changePt}(t)$ to zeros

else

(6) Apply T-MSBL on $\mathbf{y}(t)$, obtaining the estimate $\widehat{\mathbf{x}}(t)$ and saving it

(7) Set $\text{changePt}(t) = 1$

if $\text{changePt}(t-1) = 0$ **then**

(8) Set $t_0 = t$

else

if $\text{changePt}(t - h_0), \cdots, \text{changePt}(t-1)$ are all ones **then**

(9) Save $\widehat{\mathbf{x}}(t_0), \cdots, \widehat{\mathbf{x}}(t)$

(10) Set $\text{changePt}(t_0 + 1), \cdots, \text{changePt}(t)$ to zeros.

(11) Set $t_0 = t$

end if

end if

end if

end for

has superiority to other algorithms.

(4) The fourth advantage comes from T-MSBL itself. Extensive experimental results (see Chapter III) have shown that T-MSBL outperforms most existing MMV algorithms.

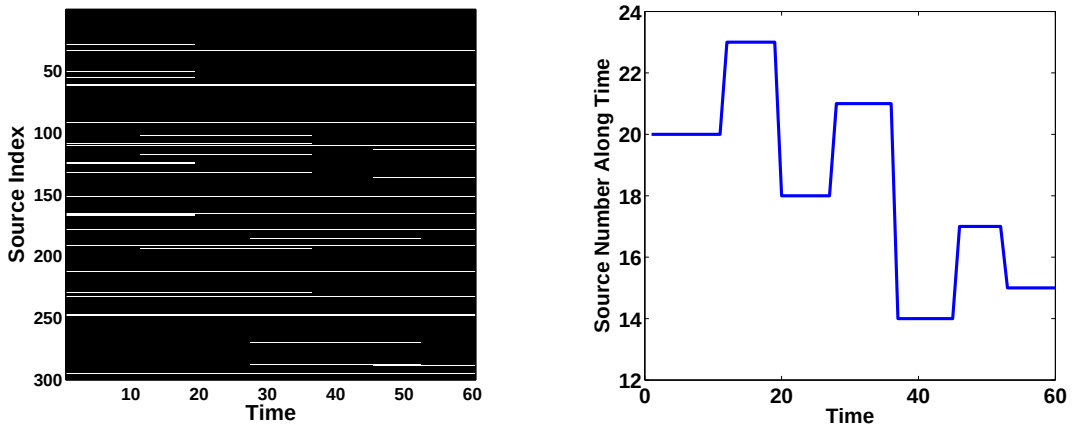
(5) Slide-TMSBL is not sensitive to the initial estimation. However, some algorithms such as LS-CS and Modified-CS require higher accuracy in the initial estimation; otherwise, these algorithms may not provide reliable estimation after some time.

(6) Slide-TMSBL can reduce the errors caused in previous estimations. To see this, suppose Slide-TMSBL has jointly estimated the source vectors from t_0 to t. Denote the estimates by $\widehat{\mathbf{X}}_{\text{pre}}$. At time $t+1$, Slide-TMSBL jointly estimates source vectors from t_0 to $t+1$ together. Denote the estimates by $\widehat{\mathbf{X}}_{\text{new}}$. If Slide-TMSBL decides that the support does not change at time $t+1$, then the estimates $\widehat{\mathbf{X}}_{\text{new}}$ will be saved as final estimates. Note that, since the estimation performance increases with increasing number of measurement vectors, the estimates $\mathbf{x}(t_0), \cdots, \mathbf{x}(t)$ in $\widehat{\mathbf{X}}_{\text{new}}$ have less estimation errors than their counterparts in $\widehat{\mathbf{X}}_{\text{pre}}$.

V.C Simulation

A computer simulation was carried out. The matrix $\mathbf{\Phi}$ was a random Gaussian matrix with the size 50×300. The number of snapshots was 60. The standard variance of noise was 0.008, which resulted in about 15 dB SNR. New sources were appeared at the 12-th, the 28-th and the 46-th snapshot. Existing sources disappeared at the 28-th and the 37-th snapshot. Figure V.1 shows the number of active sources along time.

Four algorithms were compared: the proposed Slide-TMSBL algorithm, the Segment-TMSBL algorithm [170], the Modified CS algorithm [139] with the initialization by T-MSBL, and the SOB-FOCUSS algorithm [163]. Figure V.2 shows the results averaged over 100 trials, where we can see Slide-TMSBL outperformed



(a) (b)

Figure V.1 Source activity pattern and active source number along time. In (a) each red line shows an active source. In (b) the total number of active sources is plotted as a function of the snapshot.

other algorithms.

V.D Conclusion

In this chapter, we proposed an online algorithm for the time-varying sparse model. This algorithm is based on our previously proposed T-MSBL algorithm. It automatically divides data stream into segments such that each segment satisfies a multiple measurement vector (MMV) model. Then it applies T-MSBL to each MMV model, exploiting the common sparsity property and temporal correlation in each MMV model. Thus, the algorithm largely benefits from MMV models.

V.E Acknowledgements

The text of Chapter V, in part, is currently being prepared for submission for publication of the material: Zhilin Zhang, Bhaskar D. Rao, "Sparse Bayesian Learning for Time-Varying Sparse Model". The dissertation author was a primary researcher and author of this paper.

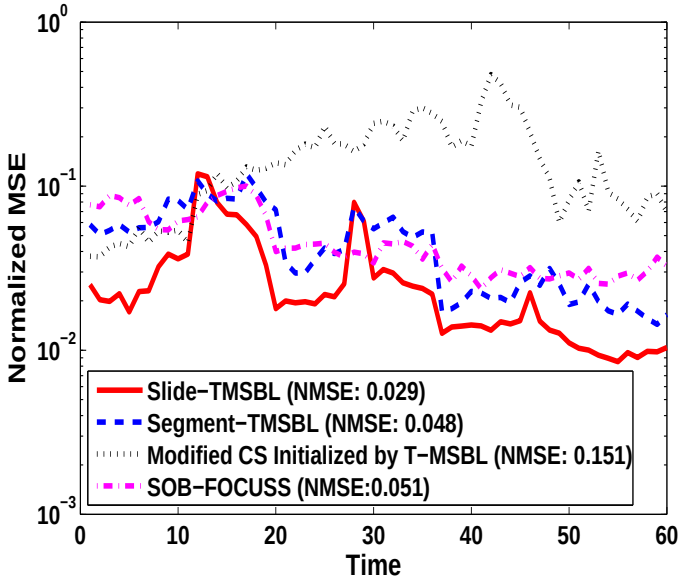


Figure V.2 Performance comparison in terms of normalized MSE at each snapshot when SNR was around 15 dB.

Chapter VI

Application: Compressed Sensing of Raw ECG Recordings for Energy-Efficient Wireless Telemonitoring

In this chapter we apply the proposed BSBL algorithms (see Chapter II) to the compressed sensing of raw ECG recordings for energy-efficient wireless telemonitoring. As an example, we mainly consider the compressed sensing of raw fetal ECG (FECG) recordings.

It is worthy emphasizing that the successes of BSBL algorithms in this application clearly show their unique ability to recover non-sparse signals, which existing compressed sensing algorithms do not have. The unique ability also has interesting mathematical meanings, which will be discussed later.

VI.A Background

Noninvasive monitoring of FECG is an important approach to monitor the health of fetuses. The characteristic parameters of an FECG, such as heart beat rates, morphology, and dynamic behaviors, can be used for diagnosis of fetal development and disease. Among these parameters, the heart beat rate is the main index of fetal assessment for high-risk pregnancies [119]. For example, abnormal patterns (decelerations, loss of high-frequency variability, and pseudo-sinusoidal) of fetal heart beat rates are generally indicative of fetal asphyxia [61].

However, noninvasive acquisition of clean FECGs from maternal abdominal recordings is not an easy problem. This is because FECGs are very weak, and often embedded in strong noise and interference, such as maternal ECGs (MECGs), instrumental noise, and artifacts caused by muscles. Further, the gestational age and the position of fetuses also affect the strength of FECGs. Up to now various signal processing and machine learning methods have been proposed to obtain FECGs, such as adaptive filtering, wavelet analysis, and blind source separation (BSS)/independent component analysis (ICA). For example, the problem of extracting clean FECGs from raw FECG recordings can be well modeled as an instantaneous ICA mixture model, in which the raw recordings are viewed as the linear mixture of a number of independent (or uncorrelated) sources includ-

ing FECG components, MECG components, and various noise components [29]. Interested readers can refer to [61, 2, 110] for good surveys on these techniques.

Traditionally, pregnant women are required to frequently visit hospitals to get resting FECG monitoring. Now, the trend and desire is to allow pregnant women to receive ambulatory monitoring of FECGs. For example, pregnant women can stay at home, where FECGs are collected through wireless telemonitoring. In such a telemonitoring system, a wireless body-area network (WBAN) [1] integrates a number of sensors attached on a patient's skin, and uses ultra-low-power short-haul radios (e.g., Bluetooth) in conjunction with nearby smart-phones or handheld devices to communicate via the Internet with the health care provider in a remote terminal. Telemonitoring is a convenient way for patients to avoid frequent hospital visits and save lots of time and medical expenses [19].

Among many constraints in WBAN-based telemonitoring systems [18], the energy consumption is a primary design constraint [93]. It is necessary to reduce energy consumption as much as possible, since a WBAN is often battery-operated. This has to be done in several ways. One way is that on-sensor computation should be minimum. Another is that data should be compressed before transmission (the compressed data will be used to reconstruct the original data in remote terminals). Unfortunately, most conventional data compression techniques such as wavelet-based algorithms dissipate lots of energy [88]. Therefore, new compression techniques are needed urgently.

Compressed sensing (CS) is a promising tool to cater to the two constraints. It uses a simple linear transform (i.e., a sensing matrix) to compress a signal, and then reconstructs it by exploiting its sparsity. The sparsity refers to the characteristics that most entries of the signal are zero. When CS is used in WBAN-based telemonitoring systems, the compression stage is completed on data acquisition module before transmission, while the reconstruction stage is completed on workstations/computers at remote receiving terminals. Based on a real-time ECG telemonitoring system, Mamaghanian et al. [88] showed that when using a sparse

binary matrix as the sensing matrix, CS can greatly extend sensor lifetime and reduce energy consumption while achieving competitive compression ratio, compared to a wavelet-based compression method. They also pointed out that when the data collection and the compression are implemented together by analog devices before analog-to-digital converter (ADC), the energy consumption can be further reduced.

Although CS has achieved some successes in adult ECG telemonitoring [88, 33], it encounters difficulties in FECG telemonitoring. These difficulties essentially come from the conflict between more strict energy constraint in FECG telemonitoring systems and non-sparsity of raw FECG recordings.

The energy constraint is more strict in FECG telemonitoring systems due to the large number of sensors deployed. Generally, the number of sensors to receive raw FECG recordings ranges from 8 to 16, and sometimes extra sensors are needed to record maternal physiological signals (e.g., blood pressure, MECG, and temperature). The large number of sensors indicates large energy dissipated in on-sensor computation. Given limited energy, this requires the systems to perform as little on-sensor computation as possible. For example, filtering before data compression may be prohibited. For CS algorithms, this means that they are required to directly compress raw FECG recordings with none or minimum preprocessing.

However, raw FECG recordings are non-sparse, which seriously deteriorates reconstruction quality of CS algorithms. Raw FECG recordings differ from adult ECG recordings in that they are unavoidably contaminated by a number of strong noise and interference, as discussed previously. Most CS algorithms have difficulty in directly reconstructing such non-sparse signals. Although some strategies have been proposed to deal with non-sparse signals, they may not be helpful in this application.

In this chapter we will apply a proposed BSBL algorithm to the compressed sensing of FECG recordings. We will see the BSBL algorithm achieves satisfactory

results.

VI.B Currently Used Models

This section discusses currently used CS models in the compressed sensing of physiological signals. As in the '*digital CS*' paradigm in [88], we assume signals have passed through the analog-to-digital converter (ADC).

The widely used CS model is the basic noiseless SMV model, expressed as

$$\mathbf{y} = \mathbf{\Phi x}, \tag{VI.1}$$

where $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is the original signal with length N. $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M \ll N)$ is a designed sensing matrix which linearly compresses $\mathbf{x}$. In this application $\mathbf{x}$ is a segment from a raw FECG recording, $\mathbf{y}$ is the compressed data which will be transmitted via a WBAN to a remote terminal.

In the remote terminal, using the designed sensing matrix $\mathbf{\Phi}$, a CS algorithm reconstructs $\mathbf{x}$ from the compressed data $\mathbf{y}$.

In many applications the signal $\mathbf{x}$ is not sparse, but sparse in some transformed domains such as the wavelet domain. This means, $\mathbf{x}$ can be expressed as $\mathbf{x} = \mathbf{\Psi\theta}$, where $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$ is an orthonormal basis matrix of a transformed domain and $\boldsymbol{\theta}$ is the representation coefficient vector which is sparse. Thus the model (VI.1) becomes

$$\mathbf{y} = \mathbf{\Phi\Psi\theta} = \mathbf{\Omega\theta}, \tag{VI.2}$$

where $\mathbf{\Omega} \triangleq \mathbf{\Phi\Psi}$. Since $\boldsymbol{\theta}$ is sparse, a CS algorithm can first reconstruct $\boldsymbol{\theta}$ using $\mathbf{y}$ and $\mathbf{\Omega}$, and then reconstruct $\mathbf{x}$ by $\mathbf{x} = \mathbf{\Psi\theta}$. This method is useful for some kinds of signals. But as shown in our experiments later, this method still cannot help existing CS algorithms to reconstruct raw FECG recordings.

Sometimes the original signal $\mathbf{x}$ itself contains noise (called '*signal noise*'). That is, $\mathbf{x} = \mathbf{u} + \mathbf{n}$, where $\mathbf{u}$ is the clean signal and $\mathbf{n}$ is the signal noise. Thus the

model (VI.1) becomes

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi}(\mathbf{u} + \mathbf{n}) = \mathbf{\Phi u} + \mathbf{\Phi n} = \mathbf{\Phi u} + \mathbf{w}, \tag{VI.3}$$

where $\mathbf{w} \triangleq \mathbf{\Phi n}$ is a new noise vector. This model can be viewed as a basic noisy SMV model.

VI.C Compressed Sensing of ECG Recordings via BSBL Algorithms

In fact, one can observe FECG recordings have rich structure. An obvious structure is the block structure as stated in Chapter II. The associated block sparse model is

$$\mathbf{y} = \mathbf{\Phi x} \tag{VI.4}$$

with

$$\mathbf{x} = [\underbrace{x_1, \cdots, x_{h_1}}_{\mathbf{x}_1^T}, \cdots, \underbrace{x_{h_{g-1}+1}, \cdots, x_{h_g}}_{\mathbf{x}_g^T}]^T \tag{VI.5}$$

A raw FECG recording can be roughly viewed as a block sparse signal contaminated by signal noise.

Figure VI.1 (a) plots a segment of a raw FECG recording. In this segment the parts from 20 to 60, from 85 to 95, and from 200 to 250 time points can be viewed as three significant non-zero blocks. Other parts can be viewed as concatenations of zero blocks. And the whole segment can be viewed as a clean signal contaminated by signal noise. Note that although the block partition can be roughly determined by observing the raw recording, it is unknown in practical FECG telemonitoring. Hence, a raw FECG recording can be modeled as a block sparse signal with unknown block partition and unknown signal noise in a noiseless environment.

Reconstructing $\mathbf{x}$ while exploiting its unknown block partition is very difficult. Up to now only several CS algorithms have been proposed for this purpose

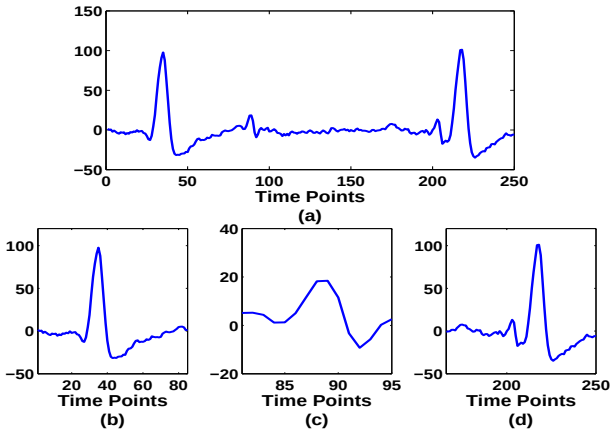


Figure VI.1 (a) A segment of an FECG recording. (b) A sub-segment containing a QRS complex of the MECG. (c) A sub-segment containing a QRS complex of the FECG. (d) A sub-segment showing a QRS complex of the FECG contaminated by a QRS complex of the MECG.

[65, 159, 101], but none of them can handle the case when the signal noise is presented.

In the following we will use a BSBL algorithm to compress/recover the FECG recordings. Particularly, we choose the BSBL-BO algorithm as an illustration (the BSBL-EM algorithm can also perform well). Although BSBL-BO needs users to define the block partition, the user-defined block partition is not needed to be the same as the true block partition. Later we will see this much clearer.

Since in wireless telemonitoring filtering and other preprocessing are not encouraged, we treat the original signal and the signal noise as a whole, i.e., compressing/recovering the signal and the signal noise together. That means, the BSBL algorithm must recover non-sparse signals. The reconstruction of non-sparse signals can be achieved by setting their γ_i-pruning threshold to a small value. The threshold is used to prune out small γ_i during learning procedures. Since in this application the signal to recover is non-sparse, we can simply set the threshold to 0, i.e., disabling the pruning mechanism.

VI.D Experiments on Real-world Datasets

Experiments were carried out using two real-world raw FECG datasets [1]. Both datasets are widely used in the FECG community. In the first dataset, the FECG is barely visible, while in the second dataset the FECG is invisible. Thus the two datasets provide a good diversity of FECG recordings to verify the efficacy of our algorithm under various situations.

For algorithm comparison, this study chooses ten representative CS algorithms. Each of them represents a family of algorithms and has top-tier performance in its family. Thus, the comparison conclusions could be generalized to other related CS algorithms.

In each experiment, all the CS algorithms used the same sensing matrix to compress FECG recordings. Thus the energy consumption of each CS algorithm was the same [2]. Therefore we only present reconstruction results.

In adult ECG telemonitoring or other applications, reconstruction performance is generally measured by comparing reconstructed recordings with original recordings using the mean square error (MSE) as a performance index. However, in our application reconstructing raw FECG recordings is not the final goal; the reconstructed recordings are further processed to extract a clean FECG by other advanced signal processing techniques such as BSS/ICA and nonlinear filtering. Due to the infidelity of MSE for structured signals [146], it is hard to see how the final FECG extraction is affected by errors in reconstructed recordings measured by MSE. Thus, a more direct measure is to compare the extracted FECG from the reconstructed recordings with the extracted one from the original recordings. This study used BSS/ICA algorithms to extract a clean FECG from reconstructed recordings and a clean FECG from original recordings, and then calculated the Pearson correlation between the two extracted FECGs.

[1] Experiment codes can be downloaded at http://dsp.ucsd.edu/~zhilin/BSBL.html , or https://sites.google.com/site/researchbyzhang/bsbl .

[2] Reconstruction of FECG recordings is done by software in remote terminals and thus it does not cost energy of WBANs.

VI.D.1 DaISy Dataset

Figure VI.1 shows a segment from the DaISy dataset [96]. Two QRS complexes of the MECG can be clearly seen from this segment, and two QRS complexes of the FECG can be seen but not very clearly. We can clearly see that the segment is far from sparse; its every entry is non-zero. This brings a difficulty to existing CS algorithms to reconstruct it.

To compress the data we used a randomly generated sparse binary sensing matrix of the size 125×250. Its each column contained 15 entries of 1s, while other entries were zero.

For the BSBL-BO algorithm, we defined its block partition according to (VI.5) with $h_1 = \cdots = h_g = 25$. Section VI.E will show that the algorithm is not sensitive to the block partition. The algorithm was employed in two ways. The first way was allowing it to adaptively learn and exploit intra-block correlation. The second way was preventing it from exploiting intra-block correlation, i.e. by fixing the matrices $\mathbf{B}_i(\forall i)$ to identity matrices.

The results are shown in Figure VI.2, from which we can see that exploiting intra-block correlation allowed the algorithm to reconstruct the segment with high quality. When the correlation was not exploited, the reconstruction quality was very poor; for example, the first QRS complex of the FECG was missing in the reconstructed segment (Figure VI.2 (c)).

Then we employed two groups of CS algorithms. One group was the algorithms based on the basic CS model (VI.1), which do not exploit block structure of signals. They were CoSaMP [99], Elastic-Net [180], Basis Pursuit [137], SL0 [94], and EM-GM-AMP [141] (with the 'heavy-tailed' mode). They are representative of greedy algorithms, of algorithms minimizing the combination of ℓ_1 and ℓ_2 norms, of algorithms minimizing ℓ_1 norm, of algorithms minimizing ℓ_0 norm, and of message passing algorithms, respectively. Note that the Basis Pursuit algorithm was the one used in [88] to reconstruct adult ECG recordings. Their reconstruction

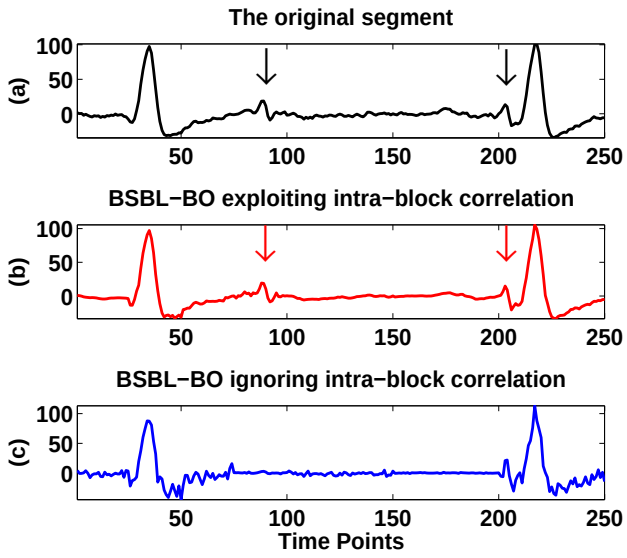


Figure VI.2 (a) The original FECG segment. (b) The reconstructed segment by BSBL-BO when exploiting intra-block correlation. (c) The reconstructed segment by BSBL-BO when not exploiting intra-block correlation. The arrows indicate QRS complexes of the FECG.

results are shown in Figure VI.3 (a)-(e) [3].

The second group was the algorithms exploiting structure of signals. They were Block-OMP [43], Block Basis Pursuit [137], CluSS-MCMC [159], StructOMP [65], and BM-MAP-OMP [101]. Block-OMP and Block Basis Pursuit need *a priori* knowledge of the block partition. We used the block partition (VI.5) with $h_1 = \cdots = h_g = h$, and h varied from 2 to 50. However, no block sizes yielded meaningful results. Figure VI.3 (f)-(g) display their results when $h = 25$. Figure VI.3 (h) shows the reconstruction result of CluSS-MCMC. StructOMP requires *a priori* knowledge of the sparsity (i.e. the number of nonzero entries in the segment). Since we did not know the sparsity exactly, we set the sparsity from 50 to 250. However, no sparsity value led to a good result. Figure VI.3 (i) shows the result with the sparsity set to 125. Figure VI.3 (j) shows the result of BM-MAP-OMP.

[3] The free parameters of these algorithms were tuned by trial and error. But no values were found to give meaningful results.

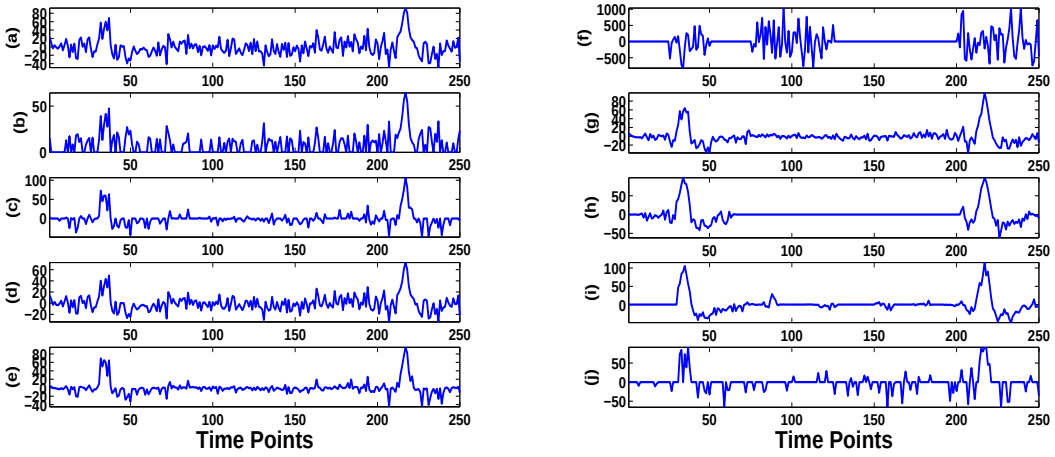


Figure VI.3 Recovery results of compared algorithms. From (a) to (j), they are the results by (a) Elastic Net, (b) CoSaMP, (c) Basis Pursuit, (d) SL0, (e) EM-GM-AMP, (f) Block-OMP, (g) Block Basis Pursuit, (h) CluSS-MCMC, (i) StructOMP, and (j) BM-MAP-OMP, respectively.

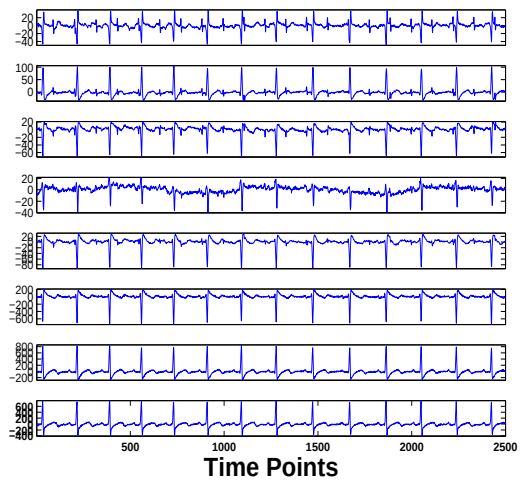
Comparing all the results we can see only the BSBL-BO algorithm, if allowed to exploit intra-block correlation, reconstructed the segment with satisfactory quality.

To further verify the ability of BSBL-BO, we used the same sensing matrix to compress the whole DaISy dataset, and then used BSBL-BO to reconstruct it.

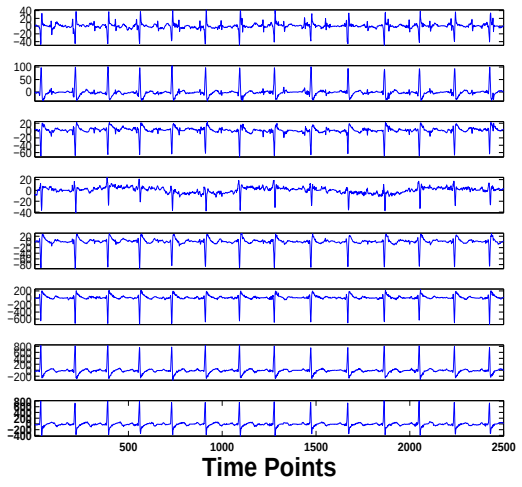
Figure VI.4 (a) shows the whole dataset. The most obvious activity is the MECG, which can be seen in all the recordings. The FECG is very weak, which is nearly discernible in the first five recordings. The fourth recording is dominated by a baseline wander probably caused by maternal respiration.

The reconstruction result by BSBL-BO is shown in Figure VI.4 (b). All the recordings were reconstructed well. Visually, we do not observe any distortions in the reconstructed dataset.

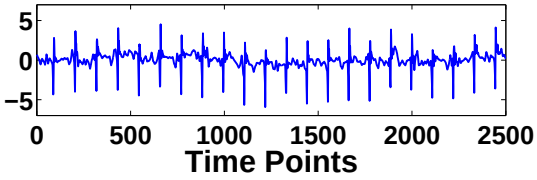
Admittedly, the reconstructed recordings contained small errors. Since the final goal in our application is to extract clean FECGs from reconstructed FECG recordings using advanced signal processing techniques such as BSS/ICA,



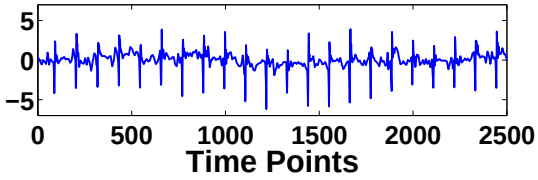
(a) Original DaISy Dataset



(b) Reconstructed Dataset



(c) FECG from (a)



(d) FECG from (b)

Figure VI.4 (a) The original dataset. (b) The reconstructed dataset by BSBL-BO. (c) The extracted FECG from the original dataset. (d) The extracted FECG from the dataset reconstructed by BSBL-BO.

we should study whether the reconstruction errors deteriorate the performance of these techniques when extracting FECGs. Here, we examined whether the errors affected the performance of BSS/ICA. We used the eigBSE algorithm, a BSS algorithm proposed in [166], to extract a clean FECG from the reconstructed recordings. The algorithm exploits quasi-periodic characteristics of FECGs. Thus, if the quasi-periodic structure of FECGs and the ICA mixing structure of the recordings are distorted, the extracted FECG will have poor quality.

Figure VI.4 (d) shows the extraction result. We can see the FECG was clearly extracted without losing any QRS complexes or containing residual noise.

For comparison, we performed the eigBSE algorithm on the original recordings to extract the FECG. The result is shown in Figure VI.4 (c). Obviously, the two extracted FECGs were almost the same. In fact, their Pearson correlation was 0.931.

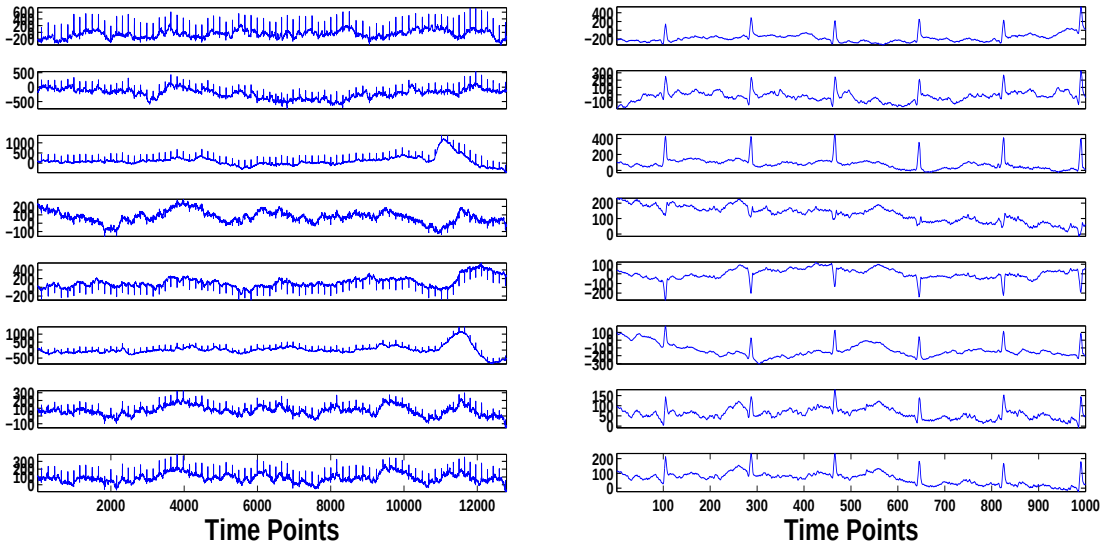
VI.D.2 OSET Database

Generally in raw FECG recordings there are many strong baseline wanders, and FECGs are very weak and are buried by noise or MECGs. To test the ability of BSBL-BO in these worse scenarios, we used the dataset 'signal01' in the Open-Source Electrophysiological Toolbox (OSET) [112]. The database consists of eight abdominal recordings sampled at 1000 Hz. We first downsampled the dataset to 250 Hz, since in WBAN-based telemonitoring the sampling frequency rarely exceeds 500 Hz. For illustration, we selected the first 12800 time points of each downsampled recording as the dataset used in our experiment. Figure VI.5 (a) shows the studied dataset, where in every recording the baseline wander is significant. Figure VI.5 (b) shows the first 1000 time points of the recordings, where the QRS complexes of the MECG and various kinds of noise dominate the recordings and the FECG is completely buried by them.

We used another randomly generated sparse binary sensing matrix of the size 256×512 with each column consisting of 12 entries of 1s with random locations, while other entries were all zero. The sensing matrix is exactly the one used in [88].

For BSBL-BO, we set the block partition $h_1 = \cdots = h_{16} = 32$. The recovered dataset by BSBL-BO is shown in Figure VI.6 (a), and the first 1000 time points of the recovered recordings are shown in Figure VI.6 (b). Visually, the recovered dataset was the same as the original dataset, even the baseline wanders were recovered well.

The previous ten CS algorithms were used to reconstruct the dataset. Again, they all failed. To save space, only the results by CluSS-MCMC and BM-



(a) Original OSET Dataset

(b) First 1000 Time Points

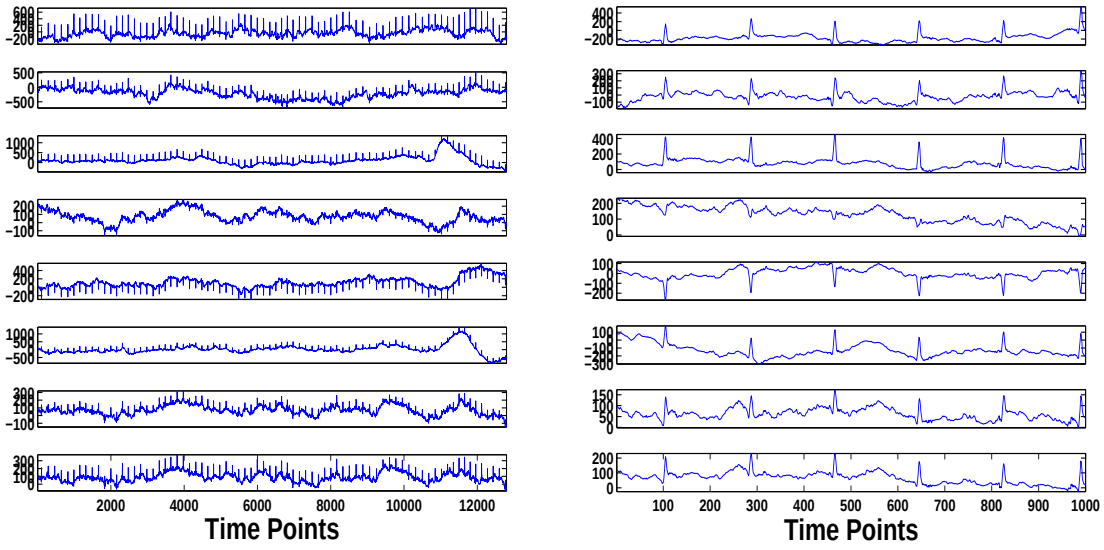
Figure VI.5 The downsampled dataset from the OSET Database. (a) The whole dataset, which contains strong baseline wanders. (b) The close-up of the first 1000 time points of the recordings, where only the QRS complexes of the MECG can be observed. The QRS complexes of the FECG are not visible.

MAP-OMP are presented (Figure VI.7).

Similar to the previous subsection, we used BSS/ICA to extract the FECG and then compared it to the one extracted from the original dataset. Here we used another ICA algorithm, the FastICA algorithm [67].

First, the reconstructed dataset was band-passed from 1.75 Hz to 100 Hz (note that in telemonitoring, it is done in the reconstruction stage in remote terminals). Then, FastICA was performed in the 'deflation' mode. Six independent components (ICs) with significant non-Gaussianity were extracted, as shown in Figure VI.8 (a), where the fourth IC is the FECG.

Then FastICA was performed on the original dataset. The ICs are shown in Figure VI.8 (b). Comparing Figure VI.8 (a) with Figure VI.8 (b) we can see the distortion was very small, which obviously did not affect clinical diagnosis.



(a) Recovered Dataset by BSBL-BO

(b) First 1000 Time Points

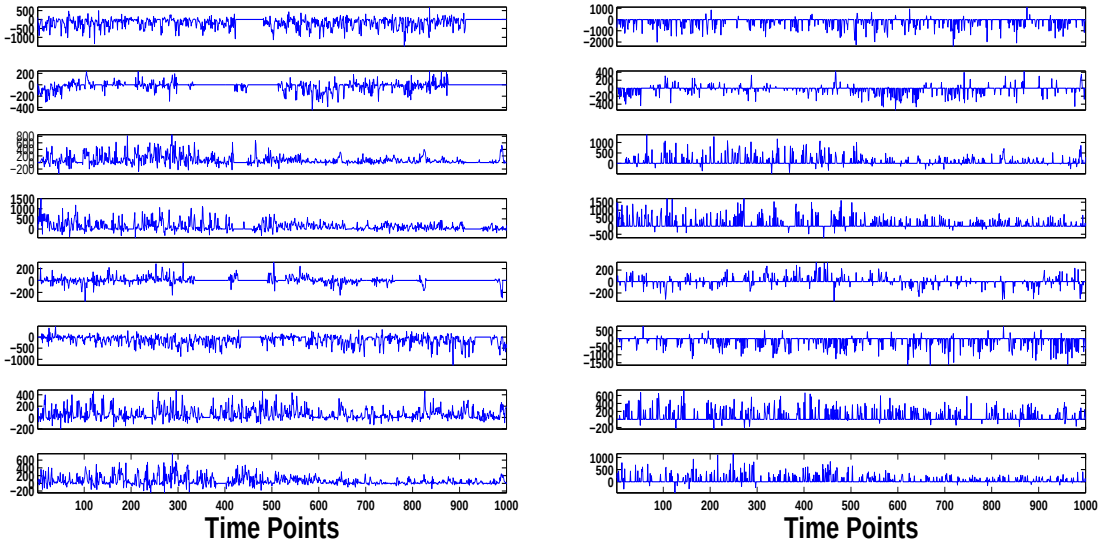
Figure VI.6 The recovered dataset by BSBL-BO. (a) The recovered whole dataset. (b) The first 1000 time points of the recovered dataset.

VI.D.3 Reconstruction in the Wavelet Domain

To reconstruct non-sparse signals, a conventional approach in the CS field is to adopt the model (VI.2), namely, first reconstructing $\boldsymbol{\theta}$ using the received data $\mathbf{y}$ and the known matrix $\boldsymbol{\Omega}$, and then calculating $\mathbf{x}$ by $\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\theta}$. To test whether this approach is helpful for existing CS algorithms to reconstruct raw FECG recordings, in the following we repeated the experiment in Section VI.D.2 using the previous ten CS algorithms and this approach.

Since it is suggested [40] that Daubechies-4 wavelet can yield very sparse representation of ECG, we set $\boldsymbol{\Psi}$ to be the orthonormal basis of Daubechies-4 wavelet. The sensing matrix was the one used in Section VI.D.2.

Unfortunately, all these CS algorithms failed again. The FECG was not extracted from the dataset reconstructed by any of these CS algorithms. Figure VI.9 (a) shows the ICs extracted from the dataset reconstructed by SL0 based on the wavelet basis. Obviously, the FECG was not extracted.



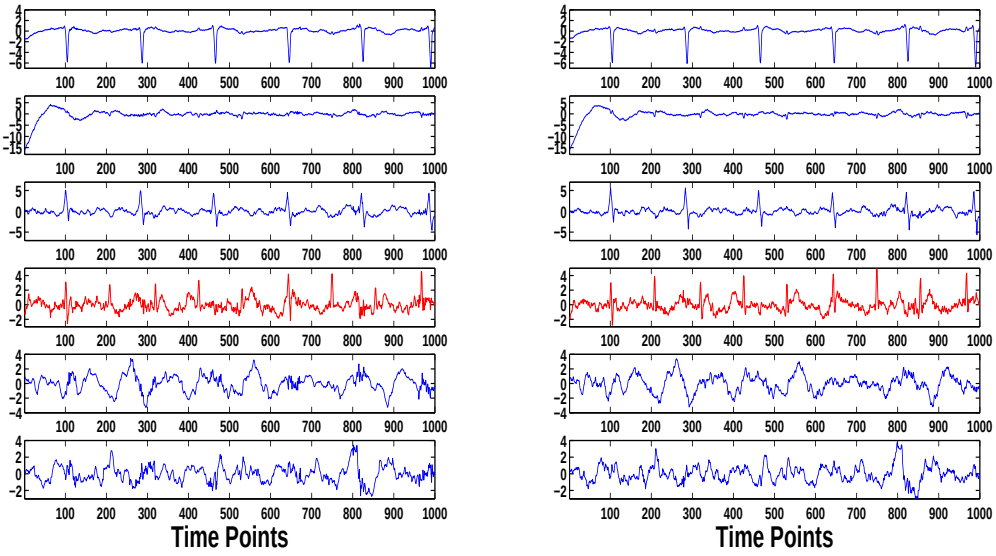
(a) Recovered by CluSS-MCMC

(b) Recovered by BM-MAP-OMP

Figure VI.7 The whole datasets recovered by (a) CluSS-MCMC and (b) BM-MAP-OMP, respectively.

Therefore, using the wavelet transform is still not helpful for these CS algorithms. The reason is that to ensure the FECG can be extracted by ICA with high fidelity, the ICA mixing structure should be maintained well in the reconstructed dataset. This requires that wavelet coefficients with small amplitudes in addition to those with large amplitudes are all recovered well. However, for a raw FECG recording the number of wavelet coefficients with small amplitudes is very large. To recover these coefficients is difficult for the CS algorithms.

As an example, the top two panels in Figure VI.9 (b) show a segment of a raw recording and its wavelet coefficients, respectively. The bottom two panels in Figure VI.9 (b) show the recovered segment and the recovered wavelet coefficients by SL0, respectively. We can see the coefficients with large amplitudes were recovered well. However, it failed to recover the coefficients with small amplitudes, which resulted in the failure of ICA to extract the FECG.



(a) ICA of the Reconstructed Dataset

(b) ICA of the Original Dataset

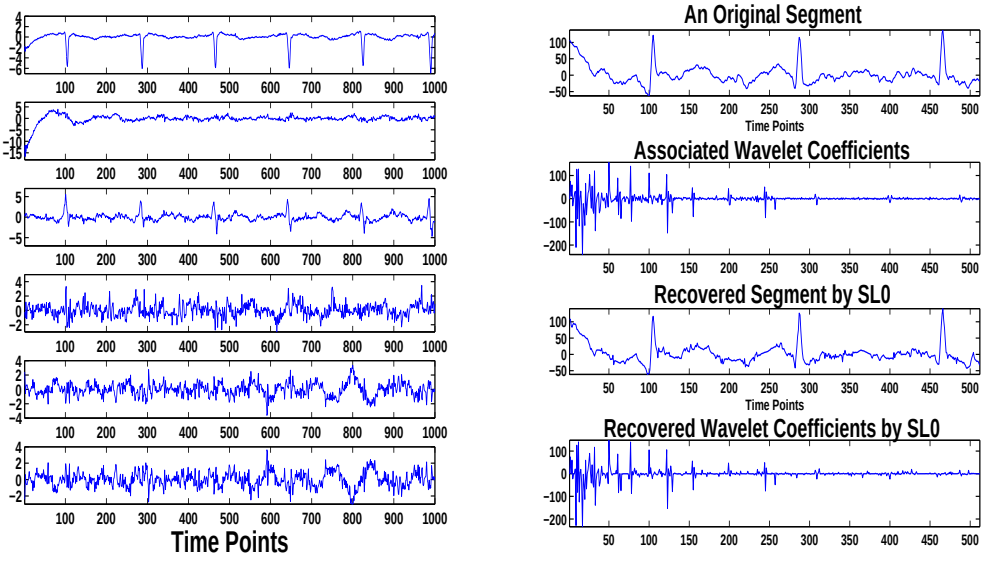
Figure VI.8 ICA decomposition on the original dataset and the recovered dataset by BSBL-BO. (a) The ICs of the recovered dataset. (b) The ICs of the original dataset. The fourth ICs in (a) and (b) are the extracted FECGs from the reconstructed dataset and the original dataset, respectively.

VI.E Performance Issues When Using BSBL Algorithms in This Application

This section explores how the performance of BSBL-BO is affected by various experimental factors.

VI.E.1 Effects of Signal-to-Interference-and-Noise Ratio

We have tested BSBL-BO's performance using two typical datasets. The two datasets contain MECGs and noise with certain strength. It is natural to ask whether BSBL-BO can be used for other datasets containing MECGs and noise with different strength. This question is very important, since different fetus positions, different pregnancy weeks, and random muscle movements can result



(a) (b)

Figure VI.9 Reconstruction result by SL0 with the aid of the wavelet transform. (a) The ICs from the recovered dataset by SL0. (b) From top to bottom are a segment of the original dataset, the associated wavelet coefficients, the recovered segment by SL0, and the recovered wavelet coefficients by SL0.

in dramatic changes in correlation structure of raw recordings, while BSBL-BO exploits the correlation structure to improve performance.

Therefore, we carried out Monte Carlo simulations with different strength of FECGs, MECGs, and other noise, as in [111]. The raw multichannel recordings were modeled as the summation of a multichannel FECG, a multichannel MECG, and multichannel noise. The multichannel MECG was generated by a three-dimentional dipole which projects cardiac potentials to eight sensors. The multichannel FECG was generated in the same way with half period of the MECG. The angle between the two dipoles generating the FECG potential and the MECG potential was 41°. The noise was a combination of randomly selected real-world baseline wanders, muscle artifacts, and electrode movement artifacts from the Noise Stress Test Database (NSTDB) [95]. Details on the simulation design can be found

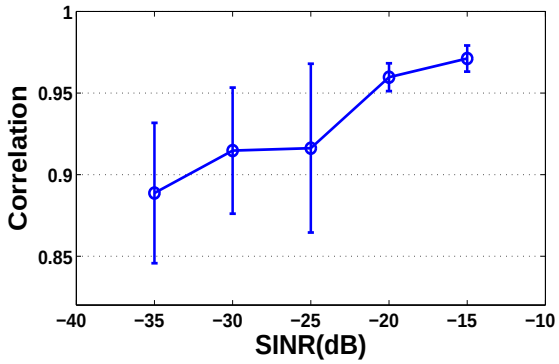
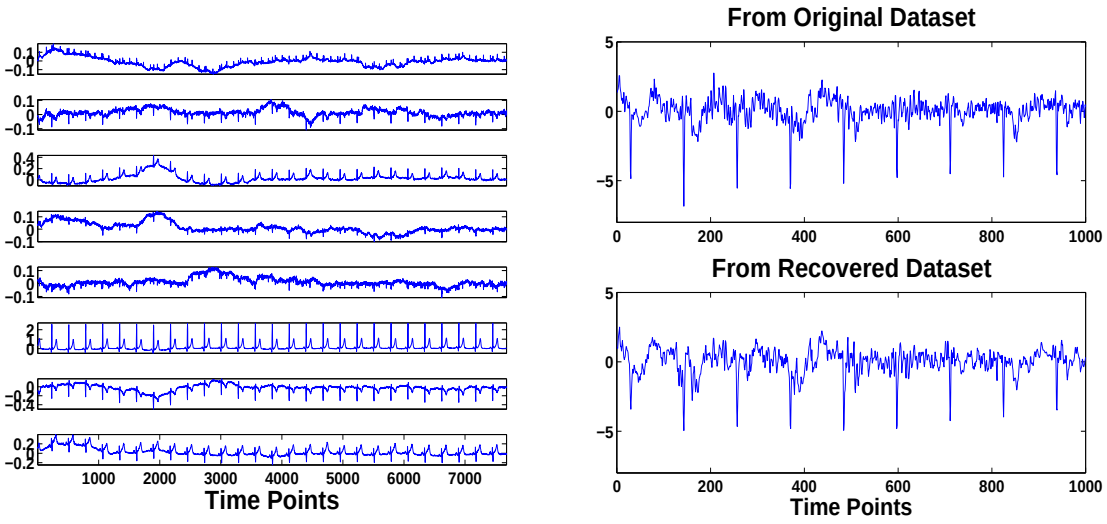


Figure VI.10 The Pearson correlation (averaged over 20 trials) between the extracted FECG from the original dataset and the one from the recovered dataset at different SINRs. The error bar gives the standard variance.

in [111, Sec. V.A]. The generated raw recordings were downsampled to 250 Hz. Each recording finally contained 7680 time points.

As in [111], the ratio of the power of the multichannel FECG to the power of the multichannel MECG was defined as the Signal-to-Interference Ratio (SIR). The ratio of the power of the multichannel FECG to the power of the multichannel noise was defined as the Signal-to-Noise Ratio (SNR). And the ratio of the power of the FECG to the combined power of the MECG and the noise was defined as the Signal-to-Interference-and-Noise Ratio (SINR). In the simulation, the strength of the FECG, the MECG and the noise were adjusted such that SNR = SIR + 10dB, and SINR was swept in the range of -35dB to -15dB. Note that in the experiment the SINR range was intentionally made more challenging, since for most raw recordings the SINR varies only from -5dB to -25dB [117]. For each value of the SINR, the simulation was repeated 20 times, each time with different signals and noise.

The sensing matrix and the block partition of BSBL-BO were the same as in Section VI.D.2. The result presented in Figure VI.10 clearly shows the high recovery quality of BSBL-BO even in the worst scenarios. Figure VI.11 shows a generated dataset when SINR=-35dB, and the extracted FECGs from the gener-



(a) (b)

Figure VI.11 A synthesized dataset and the extraction result at SINR=-35dB. (a) The synthesized dataset. (b) The comparison between the extracted FECG from the synthesized dataset and the one from the corresponding recovered dataset (only their first 1000 time points are shown).

ated dataset and from the recovered dataset. We can see the noise was very strong, but the extracted FECG from the recovered dataset still maintained high fidelity.

VI.E.2 Effects of the Block Partition

In all the previous experiments we used certain block partitions. Another question is, "Is the performance of BSBL-BO sensitive to the block partition?" To examine this, we used the dataset in Section VI.D.2. The block partition was designed as follows: the location of the first entry of each block was $1, 1+h, 1+2h, \cdots$, respectively, where the block size h ranged from 4 to 90. The sensing matrix was a sparse binary matrix of the size 128×256. Its each column contained 12 nonzero entries of 1s with random locations. The experiment was repeated 20 times. In each time the sensing matrix was different.

The averaged results are shown in Figure VI.12, from which we can see that

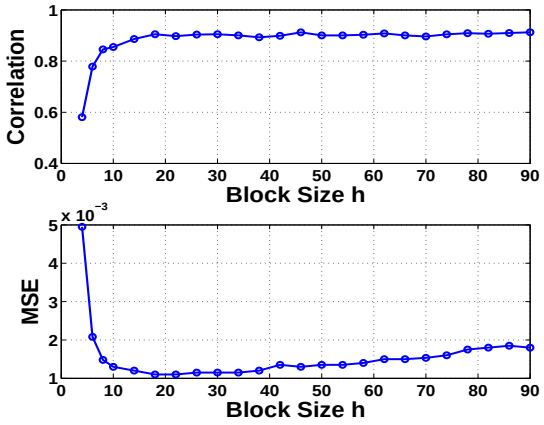


Figure VI.12 Effects of the block size h on the reconstruction quality, measured by the correlation between the extracted FECG from the reconstructed dataset and the extracted one from the original dataset (upper panel), and by the MSE of the reconstructed dataset (bottom panel).

the extraction quality was almost the same over a broad range of h.

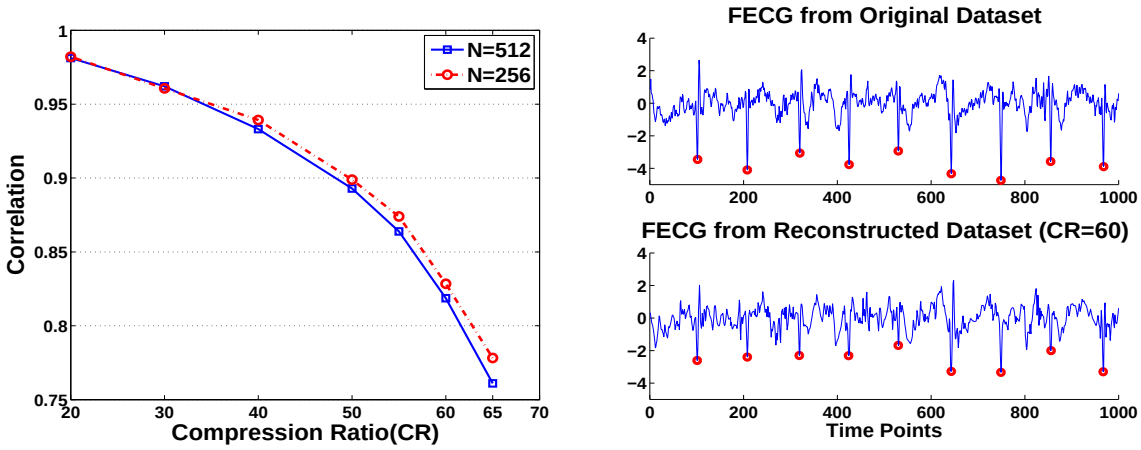
VI.E.3 Effect of Compression Ratio

Next, we investigated the effect of compression ratio (CR) on the quality of extracted FECGs from reconstructed recordings. The compression ratio is defined as

$$CR = \frac{N - M}{N} \times 100 \tag{VI.6}$$

where N is the length of the original signal and M is the length of the compressed signal. The used sparse binary sensing matrix was of the size $M \times N$, where N was fixed to 512 and M varied such that CR ranged from 20 to 65. Regardless of the size, its each column contained 12 entries of 1s. For each value of M, we repeated the experiment 20 times, and in each time the sensing matrix was randomly generated. The dataset and the block partition for BSBL-BO were the same as in Section VI.D.2.

The averaged results for each value of CR are shown in Figure VI.13 (a).



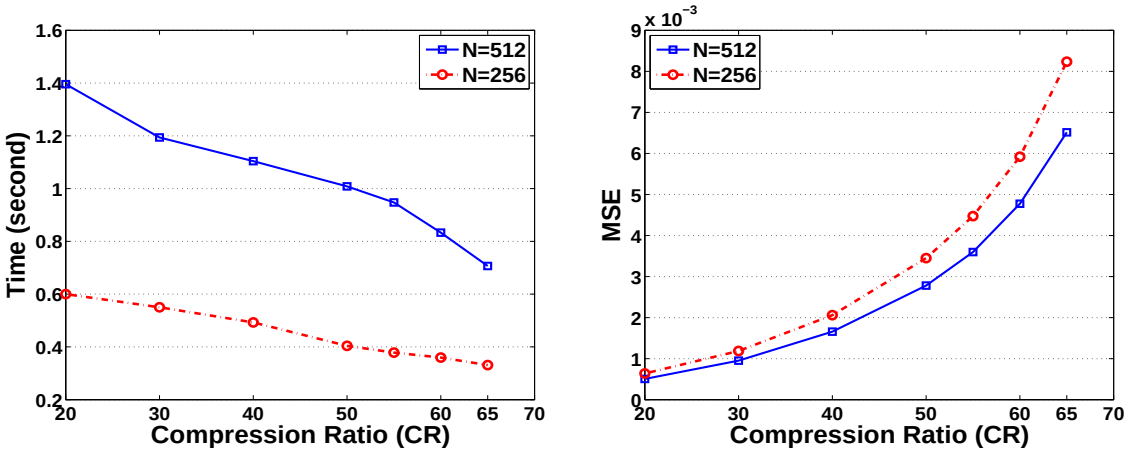
(a) (b)

Figure VI.13 (a) Effect of CR on the quality of extracted FECGs from reconstructed datasets (measured by the Pearson correlation) when $N = 512$ and $N = 256$. (b) Extracted FECG from the original dataset and from the recovered dataset when CR=60 and $N = 512$ (only first 1000 time points are shown).

We found that when CR $\leq$ 60, the quality of extracted FECGs was satisfactory and could be used for clinical diagnosis. For example, Figure VI.13 (b) shows the extracted FECG from a reconstructed dataset when CR=60. Compared to the FECG extracted from the original dataset, the FECG from the reconstructed dataset did not have significant distortion. Especially, when using the '*PeakDetection*' program in the OSET toolbox to detect peaks of R-waves in both extracted FECGs, the results were almost the same, as shown in Figure VI.13 (b), where red circles indicate the detected peaks of R-waves in both FECGs.

We repeated the experiment using a smaller sparse binary matrix with $N = 256$. Each column also contained 12 entries of 1s. The block size in the block partition for BSBL-BO did not change. The result (Figure VI.13 (a)) shows that the quality of extracted FECGs was slightly better than the case of $N = 512$.

Note that a significant advantage of using a smaller sensing matrix is that the reconstruction is accelerated. Figure VI.14 (a) compares the averaged time in reconstructing a segment of 512 time points when using two sensing matrices



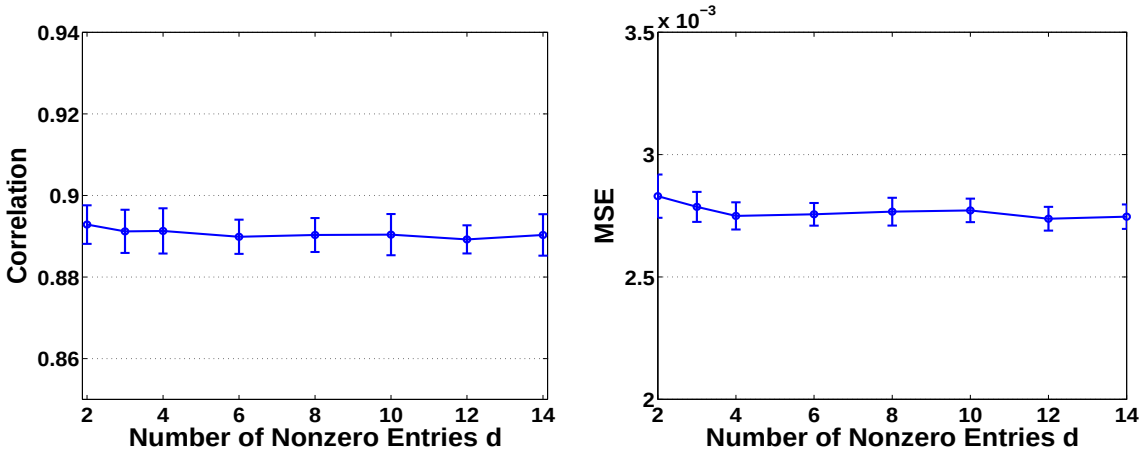
(a) (b)

Figure VI.14 (a) Comparison of averaged time in reconstructing a segment of 512 time points from the dataset shown in Figure VI.5 when using two sensing matrices ($N = 256$ and $N = 512$). (b) Comparison of MSE in reconstructing the dataset when using the two sensing matrices.

($N = 256$ and $N = 512$) at different values of CR. Clearly, using a small sensing matrix speeded up the reconstruction [4], making it possible to build a near real-time telemonitoring system. For example, BSBL-BO took less than 1.4 seconds to recover the segment in a laptop with 2.8G CPU and 6G RAM if using the big sensing matrix ($N = 512$), but took less than 0.6 seconds if using the small sensing matrix ($N = 256$).

It is worth noting that when fixing CR, using a smaller sensing matrix generally results in higher MSE of reconstructed recordings, as shown in Figure VI.14 (b). But this does not mean the quality of extracted FECGs is poorer accordingly, as shown in Figure VI.13 (a).

[4]The maximum iteration of BSBL-BO using the two sensing matrices was fixed to 25.



(a) (b)

Figure VI.15 Effect of d on recovery quality. The recovery quality is measured by the correlation between the extracted FECG from the reconstructed dataset and the FECG from the original dataset (a), and by MSE of the reconstructed dataset (b). The error bar gives the standard variance.

VI.E.4 Study on the Number of Nonzero Entries in Each Column of the Sensing Matrix

In most experiments we used a 256×512 sensing matrix, and each column contained 12 entries of 1s with random locations. This number of nonzero entries in each column was chosen by Mamaghanian et al in [88]. To study how the number of nonzero entries in each column affects the performance of BSBL-BO, we carried out a similar experiment as in [88].

The sparse binary sensing matrix was of the size 256×512. Each column contained d entries of 1s, where d varied from 2 to 14. The experiment was repeated 20 times for each value of d. In each time the locations of the nonzero entries were randomly chosen (but the generated sensing matrix was always full row-rank). The dataset and the block partition were the same as in Section VI.D.2.

Figure VI.15 (a) shows the Pearson correlation between the extracted FECG from the reconstructed dataset and the extracted FECG from the original dataset

at different values of d. Figure VI.15 (b) shows the quality of reconstructed datasets measured by the MSE. Both figures show that the results were not affected by d. This is different from the results in [88], where a basic ℓ_1 CS algorithm was used and its performance was very sensitive to d.

The robustness to d is another advantage of BSBL-BO, which is important to energy saving, as discussed in Section VI.F.3.

VI.F Further Discussions on the Use of BSBL Algorithms for this Application

VI.F.1 Block Partition in the BSBL Framework

The problem of reconstructing a raw FECG recording can be cast as a block sparse model with unknown block partition and unknown signal noise in a noiseless environment. To exploit the unknown block structure, our algorithm is based on a very simple and even counter-intuitive strategy. That is, *the user-defined block partition can be rather arbitrary, which is not required to be the same as the true block structure of the FECG recording.* This strategy is completely different from the strategies used by many CS algorithms to deal with unknown block structure, which try to find the true block structure as accurately as possible [159, 101]. In fact, the block partition in the BSBL framework is a *regularization* for better estimation of the covariance matrix of $\mathbf{x}$ in a high-dimensional parameter space. Theoretically explaining the empirical strategy in the BSBL framework is an important topic in the future.

VI.F.2 Reconstruction of Non-Sparse Signals

Most raw physiological signals are not sparse, especially when contaminated by various noise. To reconstruct these non-sparse signals, there are two popular strategies.

One is using thresholding [33] to set entries of small amplitudes to zero.

However, these thresholding methods cannot be used for FECG recordings. As we have seen, the amplitudes of FECGs are very small and even invisible. Thus it is difficult or even impossible to choose an optimal threshold value. What's worse is that the thresholding methods can destroy interdependence structure among multichannel recordings, such as the ICA mixing structure.

Another strategy widely used by CS algorithms is reconstructing signals first in transformed domains, as expressed in (VI.2). The success of this strategy strongly depends on the sparsity level of the representation coefficients $\boldsymbol{\theta}$. Unfortunately, for most raw physiological signals, the representation coefficients $\boldsymbol{\theta}$ are still not sparse enough; although coefficients of large amplitudes are few, the number of coefficients of small amplitudes is very large. When reconstructed signals are going to be further processed by other signal processing/machine learning techniques, reconstructing these coefficients of small amplitudes is important. As shown here, the failure to reconstruct these coefficients resulted in the failure of ICA to extract FECGs.

The BSBL-BO algorithm, unlike existing algorithms, directly reconstructs non-sparse signals without resorting to the above two strategies. Its reconstruction with high quality allows further signal processing or pattern recognition for clinical diagnosis. Clearly, exploiting block structure and intra-block correlation plays a crucial role in the reconstruction.

VI.F.3 Energy-Saving by the BSBL Framework

This work focuses on algorithms for wireless FECG telemonitoring. It does not involve the analysis of energy consumption, such as the comparison between BSBL-BO and wavelet compression. However, this issue actually has been addressed in the work by Mamaghanian et al. [88]. According to their 'digital CS' paradigm, if two CS algorithms use the same sensing matrix, their energy consumption is the same. Since in most experiments we used the same sparse binary matrix as theirs (12 entries of 1s in each column of $\boldsymbol{\Phi}$), their analysis on the en-

ergy consumption and their comparison between their CS algorithm and wavelet compression are applicable to BSBL-BO.

But BSBL-BO can further reduce the energy consumption while maintaining the same reconstruction performance. In Section VI.E.4 we have shown that BSBL-BO has the same performance regardless of the values of d (d is the number of entries of 1s in each column of $\mathbf{\Phi}$). Thus we can use a sparse binary sensing matrix with $d = 2$ to save more energy.

For example, when compressing a signal of 512 time points to 256 time points, using a sparse binary sensing matrix with $d = 2$ only needs about 768 additions, while using a sparse binary sensing matrix with $d = 12$ requires about 5888 additions. Thus, using the sparse binary matrix with $d = 2$ can greatly reduce code execution in CPU, thus reducing energy consumption. Note that when using a Daubechies-4 Wavelet to compress the signal, it requires 11784 multiplications and 11272 additions. In addition, the seeking of wavelet coefficients of large amplitudes also costs extra energy.

It should be noted that it seems that only BSBL-BO (and other algorithms derived from the BSBL framework) can use such a sparse binary sensing matrix with $d = 2$ to compress signals. Our experiments on adult ECGs [5] showed that other CS algorithms failed to reconstruct or had degraded reconstruction quality when using this sensing matrix. In [88] it is also shown that the basis pursuit algorithm was very sensitive to d; when d decreased from 12 to 2, the reconstruction performance measured by output SNR decreased from 20 dB to 7 dB (when the sensing matrix was of the size 256×512).

VI.F.4 Significance of the BSBL Framework

The ability of the BSBL framework to recover non-sparse signals has interesting mathematical implications. By linear algebra, there are infinite solutions to the underdetermined problem (VI.1). When the true solution $\mathbf{x}_{\text{true}}$ is sparse, using

[5] Since the compared ten CS algorithms failed to reconstruct FECG recordings, we used adult ECGs without noise in the experiments. Due to space limit the results are omitted here.

CS algorithms it is possible to find it. But when the true solution $\mathbf{x}_{\text{true}}$ is non-sparse, finding it is more challenging and new constraints/assumptions are called for. This work shows that when exploiting the block structure and the intra-block correlation of $\mathbf{x}_{\text{true}}$, it is possible to find a solution $\widehat{\mathbf{x}}$ which is very close to the true solution $\mathbf{x}_{\text{true}}$. These findings raise new and interesting possibilities for signal compression as well as theoretical questions in the subject of sparse and non-sparse signal recovery from a small number of measurements (i.e., the compressed data $\mathbf{y}$).

VI.G Conclusion

FECG telemonitoring via wireless body-area networks with low-energy constraint is a challenge for CS algorithms. This chapter showed that the block sparse Bayesian learning framework can be successfully employed in this application. Its success relies on two unique abilities; one is the ability to reconstruct non-sparse structured signals, and the other is the ability to explore and exploit correlation structure of signals to improve performance. Although the focus is the wireless FECG telemonitoring, the proposed framework and associated algorithms can be used to many other telemedicine applications, such as telemonitoring of adult ECG, wireless electroencephalogram, and electromyography.

VI.H Acknowledgements

The text of Chapter VI, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Non-Invasive Fetal ECG via Block Sparse Bayesian Learning", to appear in IEEE Trans. on Biomedical Engineering, 2013. The dissertation author was a primary researcher and author of the cited paper.

Chapter VII

Application: Compressed Sensing of Multichannel ECG Recordings for Wireless Telemonitoring via STSBL Algorithms

In the previous chapter we applied a BSBL algorithm to the compressed sensing (CS) of raw FECG recordings, and achieved successes. However, BSBL was designed for recovering single-channel signals. When recovering multichannel recordings, BSBL has to recover the signals channel by channel, which costs lots of time and thus is not suitable for real-time telemonitoring. Furthermore, for many kinds of physiological signals such as multichannel ECG recordings and multichannel EEG recordings, there is strong correlation among different channel recordings. Exploiting this correlation can greatly improve algorithms' performance. However, BSBL does not exploit it.

In this chapter we apply the spatiotemporal sparse Bayesian learning framework, developed in Chapter IV, to the CS of raw multichannel ECG recordings (including fetal ECG recordings and adult ECG recordings). It not only exploits the correlation structure in each channel signal as BSBL, but also exploits the correlation among signals of different channels. Therefore it has better recovery performance than BSBL. Besides, due to the ability to jointly recover multichannel signals, it has much faster speed than BSBL. In this sense, the proposed framework is more attractive to real-time telemonitoring of multichannel signals.

VII.A Literature Review on CS of ECG Recordings

In CS of physiological signals such as ECG and EEG, the widely used model is the basic SMV model. A lot of works have been done using this model. For example, Dixon et al. [33] compared the SMV-model-based CS framework with some conventional and adaptive sampling techniques, and considered several system-level design issues when using the CS framework. Chen et al. [20] proposed an energy-efficient digital implementation of the CS architecture for data compression in wireless sensors. Using a real-time wireless body sensor network system, Mamaghanian et al. [88] compared a basic SMV algorithm with state-of-the-art wavelet compression methods, showing that the SMV algorithm can largely save

energy and extend sensor lifetime, while achieving competitive compression ratios compared to the wavelet compression methods.

Although most work on the CS of physiological signals considered the basic SMV algorithms, recently people have noticed that the structure information in natural signals can be exploited to improve recovery quality. For example, when ECG is represented by wavelet basis functions, the wavelet coefficients have some kinds of interdependence structure. Thus several groups proposed to use tree-structure based recovery algorithms to recover ECG, which achieved better results than using basic SMV algorithms [89, 102]. In the previous chapter we proposed to use the BSBL algorithms to recover signals via exploiting correlation among successive sampling points of signals.

In addition to the SVM model, another widely used model is the MMV model. It can be expressed as follows:

$$\mathbf{Y} = \mathbf{\Phi X} + \mathbf{V}, \tag{VII.1}$$

where $\mathbf{Y} \in \mathbb{R}^{M\times L}$, $\mathbf{X} \in \mathbb{R}^{N\times L}$, and $\mathbf{V} \in \mathbb{R}^{M\times L}$. A key assumption in the MMV model is that the support of each column of $\mathbf{X}$ is identical. In [103] Polania et al. used the T-MSBL algorithm to recover single-channel ECG recordings. They first detected R peaks in an ECG recording, thus obtaining each ECG cycle. Then the cycles were normalized to identical length and formed columns of $\mathbf{X}$. Next, $\mathbf{X}$ was compressed and sent to a remote terminal for recovery by T-MSBL.

But note that a lot of works may not be suitable for energy-efficient wireless telemonitoring. This is because wireless telemonitoring has its own specific challenges [167], namely the sharp conflict between energy constraint and the non-sparsity of raw physiological signals. This has been discussed in the previous chapter.

To solve this challenge, we proposed to use the BSBL framework for CS of non-sparse physiological signals, and achieved successes. However, BSBL was designed for recovering single-channel signals. When recovering multichannel recordings, BSBL has to recover the signals channel by channel, which costs lots of

time and thus is not suitable for real-time telemonitoring. Furthermore, for many kinds of physiological signals such as multichannel ECG recordings and multichannel EEG recordings, there is strong correlation among different channel recordings. Exploiting this correlation can greatly improve algorithms' performance. However, BSBL does not exploit it.

In the following we use the STSBL framework (see Chapter IV) to jointly recover multichannel physiological signals. STSBL not only exploits the correlation structure in each channel signal as BSBL, but also exploits the correlation among signals of different channels. Therefore it has better recovery performance than BSBL. In other words, it can achieve larger compression ratio than BSBL when their recovery quality is the same. Besides, due to the ability to jointly recover multichannel signals, it has much faster speed than BSBL. Therefore, STSBL is more attractive to real-time telemonitoring of multichannel signals.

VII.B Issues When Use STSBL for this Application

Several issues should be noticed when using the STSBL framework for the CS of multichannel physiological signals. For convenience, we first write the spatiotemporal sparse model below

$$\mathbf{Y} = \mathbf{\Phi X} + \mathbf{V}, \tag{VII.2}$$

where $\mathbf{X}$ has the block structure

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{[1]\cdot} \\ \mathbf{X}_{[2]\cdot} \\ \vdots \\ \mathbf{X}_{[g]\cdot} \end{bmatrix} \tag{VII.3}$$

When applying the spatiotemporal model (VII.2) to the compressed sensing of multichannel physiological signals, the l-th column of $\mathbf{X}$ is an original signal segment in the l-th channel. The l-th column of $\mathbf{Y}$ is the corresponding compressed signal segment in this channel.

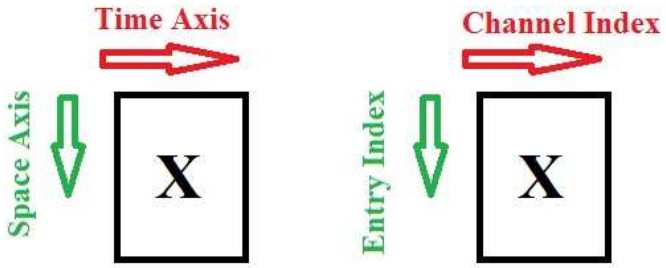


Figure VII.1 The physical meanings of the coordinates of $\mathbf{X}$ in different contexts. See the text for details.

One needs to notice the difference between the coordinate meanings of our model/algorithm and their physical meanings in the application context. As illustrated in Figure VII.1 (left), when we describe our model and the associated algorithm, the lateral axis of the matrix $\mathbf{X}$ refers to the temporal domain, while the vertical axis refers to the spatial domain [1]. When we apply the algorithm to the compressed sensing of multichannel signals, the lateral axis refers to the channel index, and the vertical axis refers to the entry index in a signal, as shown in Figure VII.1 (right).

For illustration, we choose STSBL-EM to perform all the experiments. Below we discuss some specific settings when applying STSBL-EM to the application, where the signals to recover are non-sparse.

First, notice the algorithm requires users to set the block partition (VII.3). To recover non-sparse signals, the setting of the block partition could be rather arbitrary, since the recovery performance of our algorithm is robust to the block partition. This property has been shown in [167] for BSBL algorithms. In fact, in both the BSBL model and the STSBL model the block partition is a kind of regularization, which helps estimate the covariance matrix of each column of $\mathbf{X}$.

[1] These descriptions are consistent with the majority of literature in various applications, particularly our works on the exploitation of temporal correlation [174] and the exploitation on spatial correlation [171].

Algorithm 3 STSBL-EM For Noiseless Scenarios

Input: $\mathbf{Y}$, $\mathbf{\Phi}$, and the block partition $\{d_1, \cdots, d_g\}$.

Output: $\mathbf{X}$

Initialization: $\mathbf{X}$ is assigned by the Least Square solution; $\mathbf{A}_i = \mathbf{I}_{d_i}(\forall i)$; $\gamma_i = 1(\forall i)$; $\lambda = 10^{-10}$

while not satisfy convergence criterion **do**

$\check{\mathbf{B}} \leftarrow \sum_{i=1}^{g} \gamma_i^{-1} \mathbf{X}_{[i]\cdot}^T \mathbf{A}_i^{-1} \mathbf{X}_{[i]\cdot}$

$\mathbf{B} \leftarrow \check{\mathbf{B}} / \|\check{\mathbf{B}}\|_{\mathcal{F}}$

$\boldsymbol{\mu} \leftarrow \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\mathbf{Y}\mathbf{B}^{-\frac{1}{2}}$

$\mathbf{\Sigma} \leftarrow \mathbf{\Pi} - \mathbf{\Pi}\mathbf{\Phi}^T(\lambda\mathbf{I} + \mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{\Pi}$

$\gamma_i \leftarrow \frac{1}{Ld_i} \sum_{l=1}^{L} \mathrm{Tr}\Big[\mathbf{A}_i^{-1}\big(\mathbf{\Sigma}_{[i]} + \boldsymbol{\mu}_{[i]l}\boldsymbol{\mu}_{[i]l}^T\big)\Big]$, $(\forall i)$

$\mathbf{A}_i \leftarrow \frac{1}{L} \sum_{l=1}^{L} \frac{\mathbf{\Sigma}_{[i]} + \boldsymbol{\mu}_{[i]l}\boldsymbol{\mu}_{[i]l}^T}{\gamma_i}$, $(\forall i)$ or adopting the regularization strategy in Section IV.D

$\mathbf{X} \leftarrow \boldsymbol{\mu}\mathbf{B}^{\frac{1}{2}}$

end while

In practice most SBL algorithms adopt a γ_i-pruning mechanism [174, 171, 118, 132, 154]. The mechanism forces γ_i of small values to zero, thus encouraging solutions to be sparse in the level of blocks [171], rows [174], or entries [132]. For example, in our model when γ_i was set to zero, the estimate of the block $\mathbf{X}_{[i]\cdot}$ becomes a zero block. However, since in our application the signals are non-sparse, a suitable strategy is to disable the γ_i-pruning mechanism.

Since in our application the noise $\mathbf{V}$ can be ignored, the parameter λ can be simply set to a very small value. In our experiments we set $\lambda = 10^{-10}$. And to improve the estimation robustness for $\mathbf{B}$, we remove the second term in (IV.23).

Algorithm 3 summarizes the STSBL-EM algorithm when used in noiseless scenarios.

VII.C Experiments on Multichannel Fetal ECG Recordings

In the previous chapter ten state-of-the-art CS algorithms based on the SMV model were performed and all failed; only BSBL-BO succeeded. Thus, this study puts emphasis on the comparison between BSBL-BO and STSBL-EM, and the comparison between STSBL-EM and CS algorithms based on the MMV model. CS algorithms based on the SMV model are not compared any more. The details of the compared algorithms are as follows.

- The BSBL-BO algorithm [171], which was used in [167] for compressed sensing of single-channel fetal ECG.
- The Champagne algorithm [152], which is a SBL algorithm. It does not exploits the temporal correlation and has limited ability to exploit the spatial correlation.
- The ISL0 algorithm [66], which is an MMV algorithm smoothly minimizing the penalty $\sum_{i=1}^{M} \mathcal{I}(\|\mathbf{X}_{i\cdot}\|_2)$ where $\mathcal{I}(a) = 1$ if $a \neq 0$, or $\mathcal{I}(a) = 0$ if $a = 0$. It is based on the assumption that $\mathbf{X}$ is row-sparse (i.e., only a few rows of $\mathbf{X}$ are nonzero).
- The SA-MUSIC algorithm [77], which is a greedy MMV algorithm. As ISL0, it assumes that $\mathbf{X}$ is row-sparse. It requires users to determine how many nonzero rows in $\mathbf{X}$. But this is impossible for our applications. So we set this number to each integer ranging from $N/4$ to N, and only reported the one associated with the smallest measure square error in the estimate of $\mathbf{X}$.

For the three SBL algorithms, i.e., STSBL-EM, BSBL-BO, and Champagne, λ was set to 10^{-10}, and their γ_i-pruning mechanisms were all disabled.

Since ISL0 and SA-MUSIC both assume that $\mathbf{X}$ is row-sparse, while in our applications $\mathbf{X}$ is not row-sparse, they are not suitable to recover signals directly

according to the MMV model. Alternatively, we considered the following recovery model:

$$\mathbf{Y} = \mathbf{\Omega Z} + \mathbf{V} \tag{VII.4}$$

where $\mathbf{\Omega} \triangleq \mathbf{\Phi D}$. $\mathbf{D}$ is a dictionary matrix such that each column of $\mathbf{X}$ can be sparsely represented under $\mathbf{D}$. That is to say, for all l, $\mathbf{X}_{.l} = \mathbf{D}\mathbf{Z}_{.l}$ where $\mathbf{Z}_{.l}$ is a sparse vector. If each column of $\mathbf{X}$ has the similar waveform, under suitably selected dictionary matrix $\mathbf{D}$, the nonzero rows of $\mathbf{Z}$ would not be too many. Therefore, to recover $\mathbf{X}$, ISL0 and SA-MUSIC first recovered $\mathbf{Z}$ and then obtained the estimate of $\mathbf{X}$ by $\mathbf{X} = \mathbf{DZ}$. In our experiments $\mathbf{D}$ was formed by the orthonormal basis of various kinds of wavelets and DCT. Besides, since in our applications $\mathbf{V}$ could be ignored, they were performed in the noiseless mode.

To measure the recovery performance, we considered three performance indexes. One is *the empirical mean square error (EMSE)*, defined as EMSE $= \frac{1}{CT}\sum_{c=1}^{C}\sum_{t=1}^{T}\|\widehat{\mathbf{S}}_{[t]c} - \mathbf{S}_{[t]c}\|_2^2/\|\mathbf{S}_{[t]c}\|_2^2$, where $\widehat{\mathbf{S}}_{[t]c}$ was the estimate of $\mathbf{S}_{[t]c}$ at the t-th epoch and c-th channel. Note that each $\mathbf{S}_{[t]c}$ ($\forall t$, $\forall c$) corresponds to the matrix $\mathbf{X}$ in the MMV model or the STSBL model.

The second index, used for the CS of fetal ECG, is *the Pearson correlation* between the extracted fetal ECG from the recovered dataset and the extracted one from the original dataset by using the same independent component analysis (ICA) algorithm with the same initialization. This performance index was proposed in [167, 168] to better detect small recovery errors for structured signals.

The third index is *the speed*, measured as the averaged running time for recovering an $\mathbf{S}_{[t]c}$ ($\forall t$, $\forall c$) on a computer with 2.8 GHz CPU and 6 G RAM.

VII.C.1 The OSET Fetal ECG Database

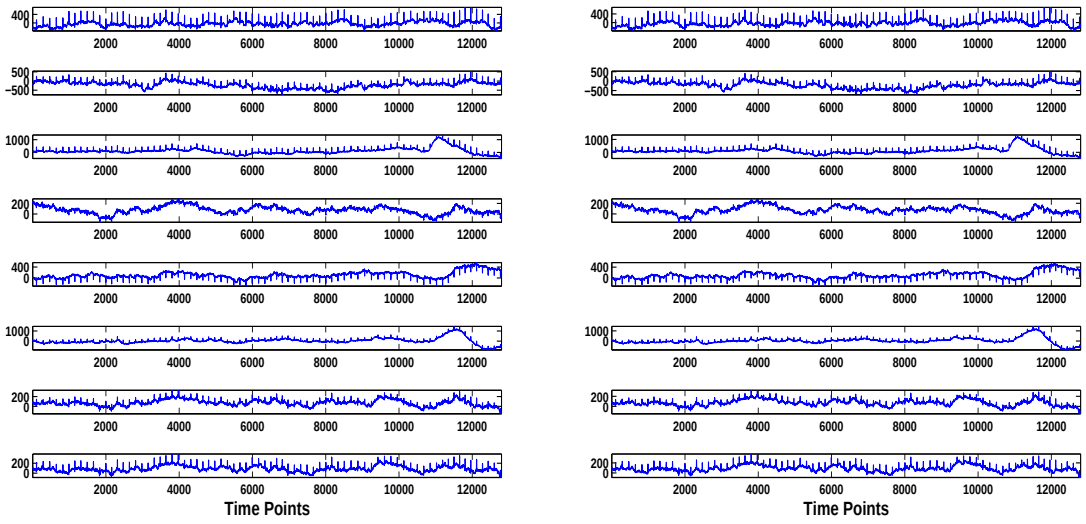
As stated in [167], recovery of raw fetal ECG recordings is extremely difficult for current CS algorithms. This is because the raw recordings are non-sparse and are contaminated by strong noise and artifacts, while the energy constraint of telemonitoring systems require little preprocessing on the raw recordings.

Thus, we first evaluate the performance of all the algorithms on typical raw multichannel fetal ECG datasets. Here we used the dataset 'signal01' in the Open-Source Electrophysiological Toolbox (OSET) [112]. It consists of eight abdominal recordings sampled at 1000 Hz. We first downsampled the dataset to 250 Hz, since in wireless telemonitoring the sampling frequency rarely exceeds 500 Hz. For illustration, we selected the first 12800 time points of each of the downsampled eight-channel recordings to form the dataset for our experiment. This dataset was the exact one used in our previous work [167]. Figure VII.2 (a) shows the dataset. Clearly, the eight-channel signals contain strong noise and artifacts. The spikes in each signal are the QRS complexes of the maternal ECG. The fetal ECG is invisible. Thus, to extract the weak fetal ECG, one needs to use ICA or other signal processing approaches. This requires the recovery quality is high. Otherwise, the extracted fetal ECG is distorted.

In [168] BSBL-BO was used for compressed sensing of the dataset. However, due to the mathematical model from which BSBL-BO is derived, it had to recover each channel signal one by one. In other words, it could not jointly recover the multichannel signals at the same time. Thus, during the recovery stage, the BSBL-BO algorithm was used to recover the signals channel by channel.

Here we applied STSBL-EM, which recovered the multichannel signals jointly. The sensing matrix $\mathbf{\Phi}$ was a sparse binary sensing matrix of the size 256×512. Its each column consisted of 12 entries of 1s with random locations, while other entries were zero. The sensing matrix is exactly the one used in [171, 88]. The block partition was $\{d_1 = \cdots = d_{32} = 16\}$. Its maximum iterations were set to 25. Figure VII.2 (b) shows the recovered dataset, which was visually the same as the original dataset.

To see the importance of exploiting spatiotemporal correlation, STSBL-EM was performed again but without exploiting spatiotemporal correlation (i.e., we set $\mathbf{B} = \mathbf{I}$ and $\mathbf{A}_i = \mathbf{I}, \forall i$). The recovered dataset is shown in Figure VII.3 (a). Clearly, the quality was poor if not exploiting spatiotemporal correlation.



(a) Original Dataset

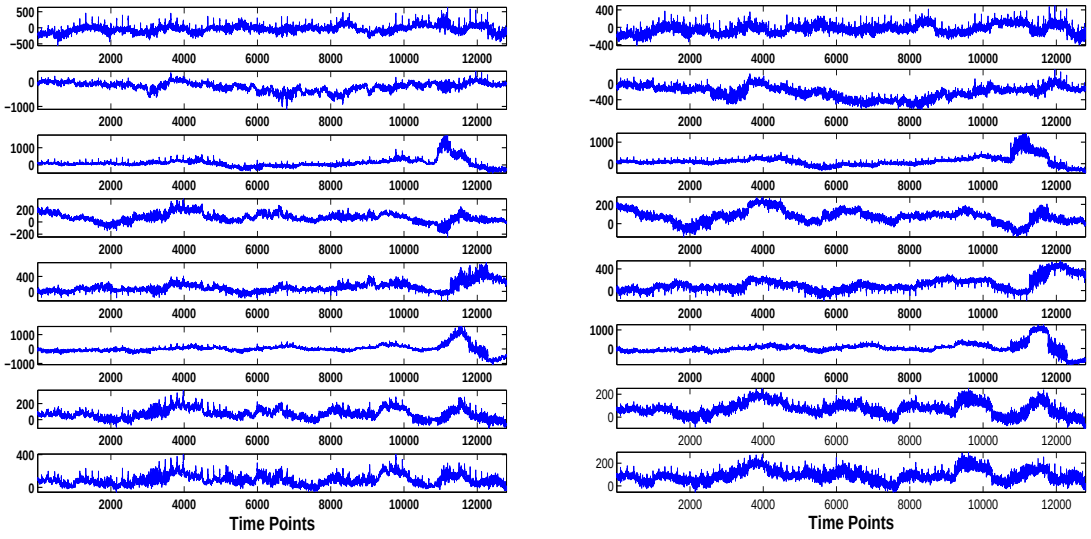
(b) Recovered by STSBL-EM

Figure VII.2 Comparison between the original dataset and the recovered dataset by STSBL-EM. (a) The original dataset. (b) The recovered dataset.

Then, we performed the Champagne algorithm. The result is shown in Figure VII.3 (b). The recovery quality was poor as well. This is probably due to its limited ability to exploit spatial correlation and complete ignorance of the interdependence among signals of different channels.

To further examine the recovery quality of our algorithm, similar as [167], we used the FastICA algorithm [67] to extract the fetal ECG from the recovered dataset, and then compared it to the one extracted from the original dataset.

First, the recovered dataset was band-passed from 1.75 Hz to 100 Hz. Then, FastICA was performed in the 'deflation' mode. Five independent components (ICs) with significant non-Gaussianity were extracted, as shown in Figure VII.4 (b), where the fourth IC is the extracted fetal ECG. Then the FastICA was performed on the original dataset. The ICs are shown in Figure VII.4 (a). Comparing the two ICA decompositions in Figure VII.4 (a) and (b), we can see the ICA decomposition on the recovered dataset had high fidelity, showing the recovery quality of STSBL-



(a) By STSBL-EM ignoring correlation

(b) By Champagne

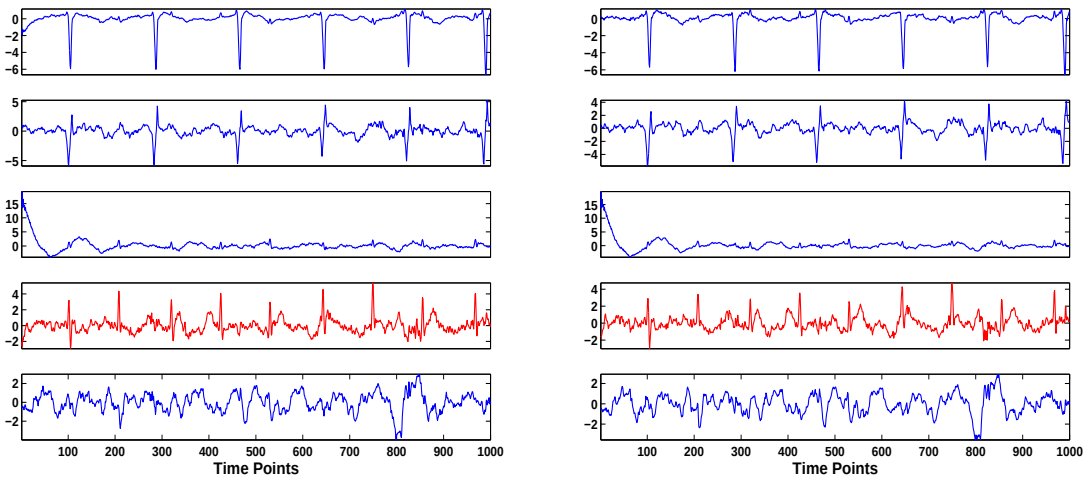
Figure VII.3 (a) The recovered dataset by STSBL-EM without exploiting spatiotemporal correlation and (b) the recovered dataset by Champagne.

EM was satisfactory.

The same procedure was performed on the recovered datasets by STSBL-EM without exploiting spatiotemporal correlation and the Champagne algorithm, but the fetal ECG was not extracted (the results are omitted).

Next, ISL0 and SA-MUSIC were used to recover the dataset employing the model (VII.4), where $\mathbf{D}$ was formed by the orthonormal basis of the Daubechies-4 wavelet. Then FastICA was performed on the recovered datasets. The extracted ICs by SA-MUSIC and ISL0 are shown in Figure VII.5 (a) and (b), respectively. One can see the two ICA decompositions were significantly distorted, and the fetal ECG was not extracted.

As shown in [167], for this dataset BSBL-BO also achieved successes. Thus the following compared BSBL-BO with STSBL-EM in terms of recovery quality (of extracted fetal ECGs) and speed at different values of compression ratio (CR).



(a) ICA of the Original Dataset

(b) ICA after STSBL-EM

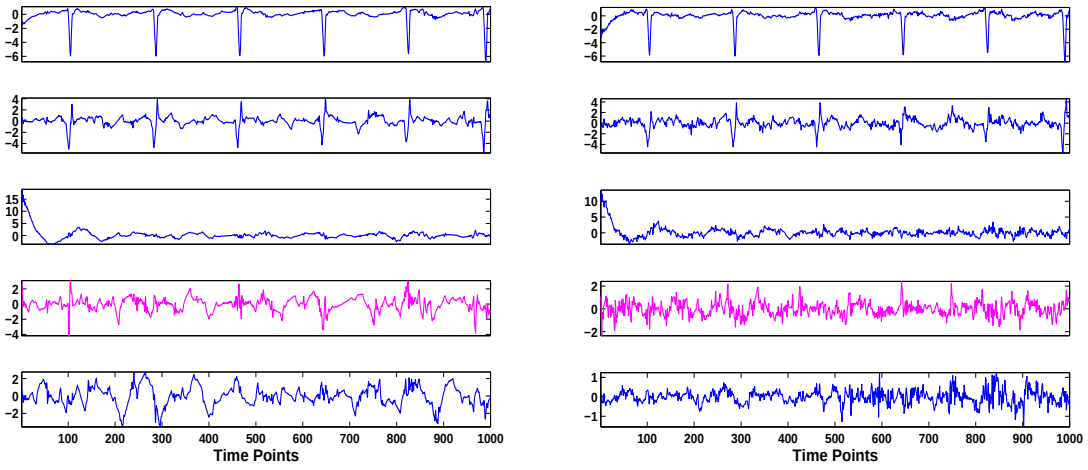
Figure VII.4 (a) ICA decomposition of the original dataset. (b) ICA decomposition of the recovered dataset by STSBL-EM. The fourth ICs indicated by the red color are the extracted fetal ECGs. Visually, there was no difference between the two ICA decompositions.

The CR is defined as

$$CR = \frac{M - N}{M} \times 100. \tag{VII.5}$$

The used sparse binary sensing matrix $\mathbf{\Phi}$ was of the size $N \times M$ with M fixed to 256, while N varied such that the CR ranged from 20 to 70. Regardless of the size, each column of $\mathbf{\Phi}$ contained 12 entries of 1s with random locations. For each value of N, we repeated the experiment 20 trials. In each trial the sensing matrix was generated again. The block partition for both algorithms was $\{d_1 = \cdots = d_{16} = 16\}$. The maximum iterations of both algorithms were set to 25.

Figure VII.6 (a) shows the Pearson correlation for both algorithms at different values of CR. Clearly, STSBL-EM outperformed BSBL-BO in all the CR range, especially at larger CR. This illustrates the benefit of exploiting the interdependence among signals of different channels. Figure VII.6 (b) shows the averaged running time of both algorithms recovering the eight-channel signals of an epoch



(a) ICA after SA-MUSIC on Wavelet

(b) ICA after ISL0 on Wavelet

Figure VII.5 (a) ICA decomposition of the recovered dataset by SA-MUSIC. (b) ICA decomposition of the recovered dataset by ISL0. Both algorithms first recovered the wavelet coefficients and then recovered the original dataset.

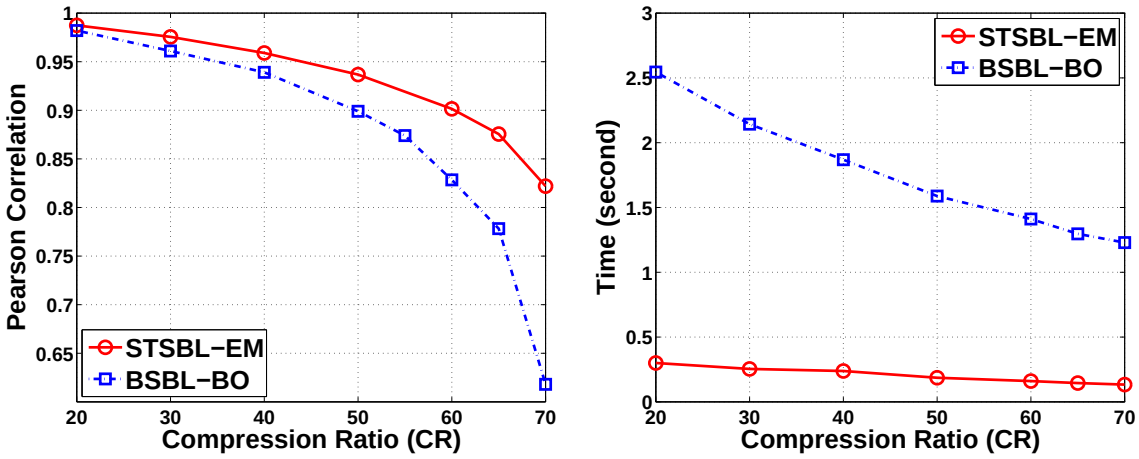
(i.e., 1.024 seconds). The speed of STSBL-EM was about eight times faster than BSBL-BO. This is because STSBL-EM jointly recovered the eight-channel signals, while BSBL-BO had to recover the signals channel by channel [2].

We repeated the same experiment on another dataset, i.e. the dataset 'signal02' in OSET. All the experiment settings were the same as before. The results are shown in Figure VII.7. Again, STSBL-EM achieved better recover quality and had much faster recovery speed.

VII.C.2 The Abdominal and Direct Fetal ECG Database

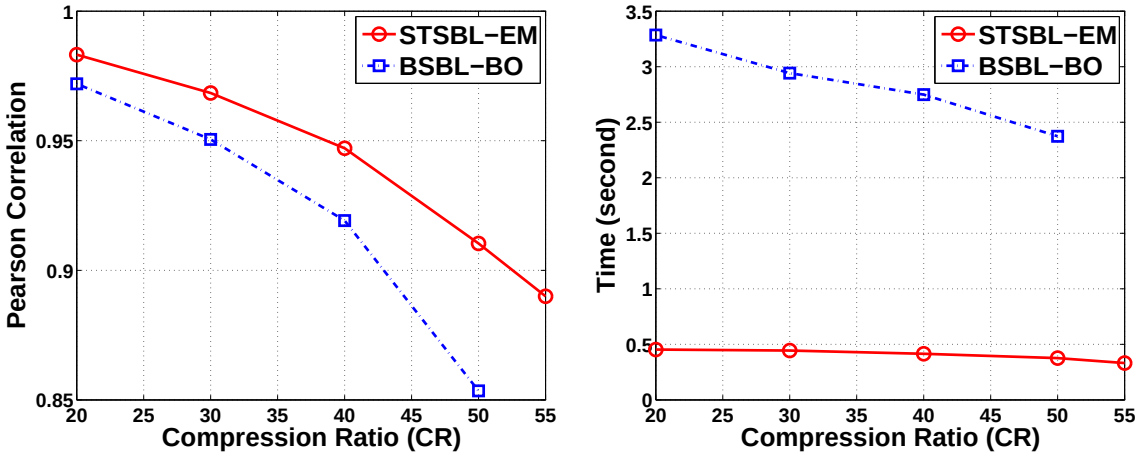
In the CS community a widely used approach to recover non-sparse signals is to resort to a dictionary matrix, as described in the model (VII.4). However, we argue that this approach so far is not effective for recovering raw physiological signals in telemonitoring scenarios. With this goal, we performed SA-MUSIC and

[2]Note that the running time of BSBL-BO in [167] was calculated on the recovery of a single-channel signal of an epoch.



(a) (b)

Figure VII.6 Effects of CR on (a) quality of extracted fetal ECGs from reconstructed datasets, and on (b) recovery time. The results are obtained on the dataset 'signal01'.



(a) (b)

Figure VII.7 Effects of CR on (a) quality of extracted fetal ECGs from reconstructed datasets, and on (b) recovery time. The results are obtained on the dataset 'signal02'. When CR = 55, the fetal ECG could not be extracted from the recovered dataset by BSBL-BO. Thus, we only plot its results when CR = $20 \sim 50$.

ISL0 to recover the compressed signals according to the model (VII.4), where five types of dictionary matrices $\mathbf{D}$ were considered, namely $\mathbf{D}$ was formed by the orthonormal basis of the Daubechies-4 wavelet, the Daubechies-12 wavelet, the Symmlet-8 wavelet, the Coiflet-4 wavelet, and the DCT. The sensing matrix $\mathbf{\Phi}$ was the same as the previous experiment. For comparison, our algorithm STSBL-EM was also performed with the same settings as before.

Since we have seen SA-MUSIC and ISL0 had poor performance on the datasets in the OSET Database, we changed to use another dataset. We performed the two algorithms on the 'r04-edfm' dataset in the Abdominal and Direct Fetal ECG Database [54]. This dataset contains recordings of six channels, but only four recordings are abdominal recordings. Since in typical telemonitoring scenarios only the abdominal recordings are available, we used the four abdominal recordings. As before, we downsampled the recordings to 250 Hz. For illustration, we selected the four recordings of the first 20.48 seconds to form the dataset for our experiment. Figure VII.8 shows the four recordings of the first 2500 time points. Clearly, the dataset contains strong noise and artifacts.

We performed SA-MUSIC, ISL0, and STSBL-EM at different values of CR (ranging from 20 to 70). At each value of CR, the experiment repeated 10 independent trials. The averaged EMSE of each algorithm is shown in Figure VII.9.

Again we see STSBL-EM significantly outperformed SA-MUSIC and ISL0. We observe that all of the used dictionary matrices were not helpful for SA-MUSIC and ISL0 to achieve high recovery performance. This is due to the fact that these raw recordings cannot be perfectly sparsely represented in the wavelet domain and the DCT domain. As stated in [167], although there are only a few coefficients of very large values in the wavelet domain, there are many wavelet coefficients of small values. Recovering these coefficients of small values is very important to the recovery quality.

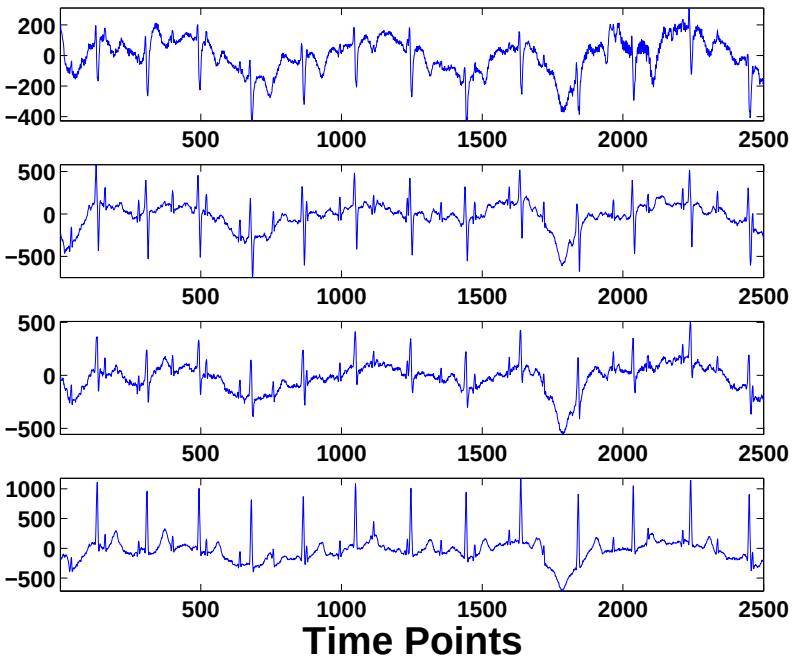


Figure VII.8 Used dataset of the first 2500 time points, which is downsampled from the dataset 'r04-edfm'. The large peaks are QRS complexes of the maternal ECG, while small peaks are QRS complexes of the fetal ECG.

VII.D Experiments on Multichannel ECG Recordings with Atrial Fibrillation

As a final example, we evaluated algorithms' performance on the dataset '04908' in the MIT-BIH Atrial Fibrillation Database [54]. This dataset contains ECG recordings of two channels, sampled at 250Hz. For illustration, in this experiment only 10240 time points of each recording were used, where the characteristics of atrial fibrillation was clearly presented (i.e., absence of P waves and irregular R-R intervals). Figure VII.10 shows the first 2000 time points of each recording.

Similar as the previous experiment, SA-MUSIC and ISL0 recovered the dataset using the Daubechies-4 wavelet, the Symmlet-8 wavelet, the Coiflet-4 wavelet, and DCT, while STSBL-EM recovered the dataset directly. The sens-

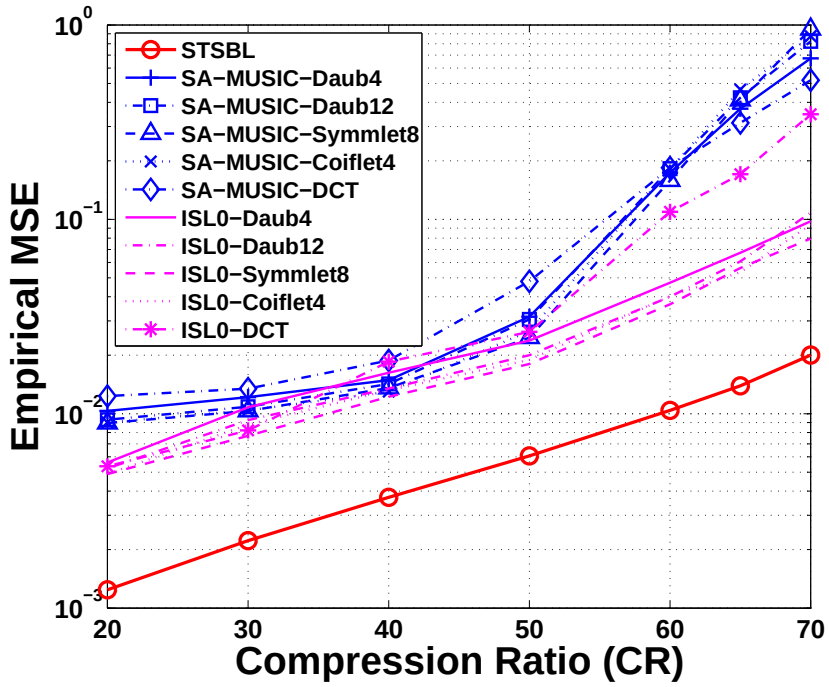


Figure VII.9 Recovery quality (measured in terms of empirical MSE) of STSBL-EM, SA-MUSIC, and ISLO. Note that SA-MUSIC and ISL0 recovered the dataset via the model (VII.4).

ing matrix was the same as the previous experiment. Note that the dataset is very clean with little noise, which is a favorite scenario of SA-MUSIC and ISL0. However, the results in Figure VII.11 show that STSBL-EM still had much better performance than the compared algorithms.

Furthermore, Figure VII.12 and Figure VII.13 show the recovered recordings by STSBL-EM and ISL0 (using the Symmlet-8 wavelet) at CR=70, respectively. Clearly, at this compression ratio, the recovery quality of STSBL-EM is still satisfactory; the recovered recordings can be used for diagnosis of atrial fibrillation. In contrast, the recovered recordings by ISL0 cannot be used for diagnosis, since there are lots of artifacts in the recordings and it is not clear whether the P waves, an important characteristics used for the diagnosis, exist or not.

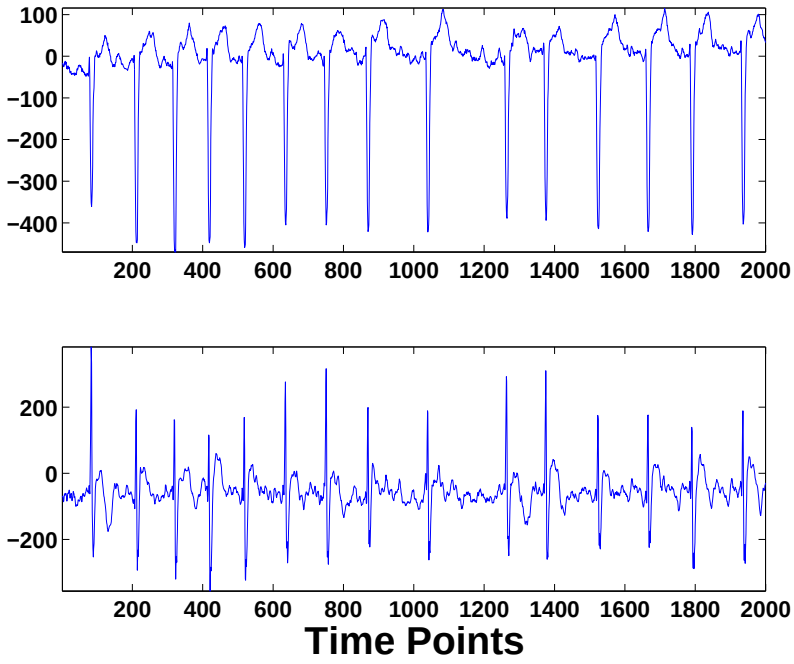


Figure VII.10 Used dataset of the first 2000 time points. The characteristics of atrial fibrillation, such as absence of P waves and irregular R-R intervals, is clearly presented.

VII.E Conclusion

In the previous chapter we applied BSBL algorithms to the compressed sensing of physiological signals. However, BSBL algorithms are designed for recovering single-channel signals, and cannot recover multichannel signals simultaneously. Thus, it may not be used in some real-time wireless telemonitoring systems, especially when the channel number is large. In this chapter, we applied a spatiotemporal sparse Bayesian learning algorithm (proposed in Chapter IV) for the compressed sensing of multichannel non-sparse physiological signals. Different from current compressed sensing algorithms, it not only exploits the correlation structure in a signal itself, but also exploits the correlation structure among signals of different channels. Experimental results showed that it not only has the best recovery performance but also has much faster speed than BSBL algorithms. Al-

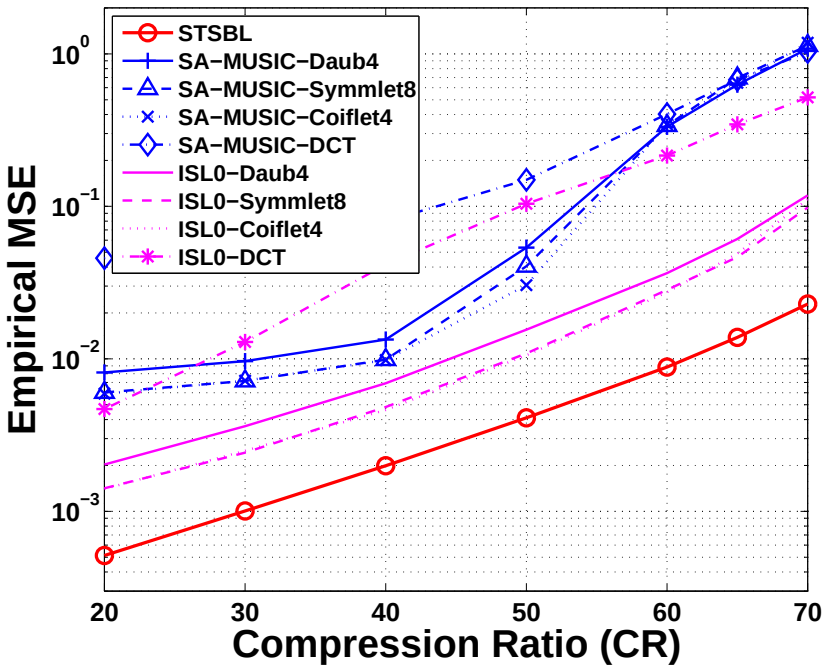


Figure VII.11 Recovery quality (measured in terms of empirical MSE) of STSBL-EM, SA-MUSIC, and ISLO. SA-MUSIC and ISL0 recovered the dataset via the model (VII.4).

though in this chapter the algorithm was only applied to the compressed sensing of multichannel ECG recordings, it can also be used to telemonitoring of other multichannel physiological signals and data collection in wireless sensor networks.

VII.F Acknowledgements

The text of Chapter VII, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Spatiotemporal Sparse Bayesian Learning with Applications to Compressed Sensing of Multichannel ECG for Wireless Telemonitoring", submitted to IEEE Trans. on Biomedical Engineering, 2012. The dissertation author was a primary researcher and author of the cited paper.

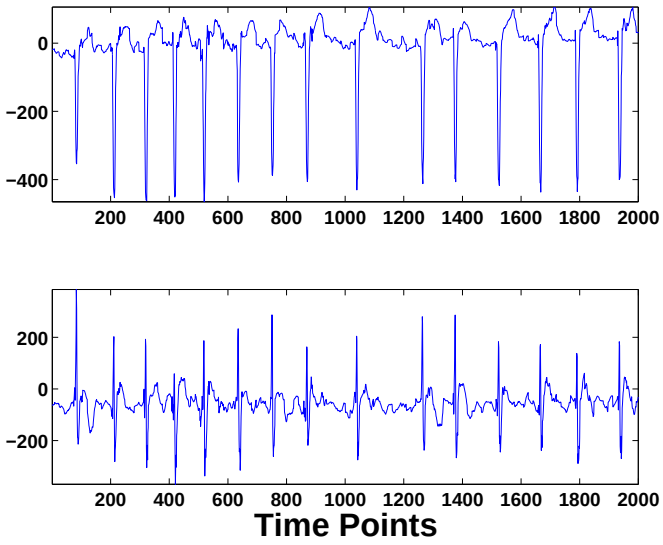


Figure VII.12 Recovered ECG recordings by STSBL-EM at CR=70. The recovered recordings by STSBL-EM can be used for diagnosis of atrial fibrillation.

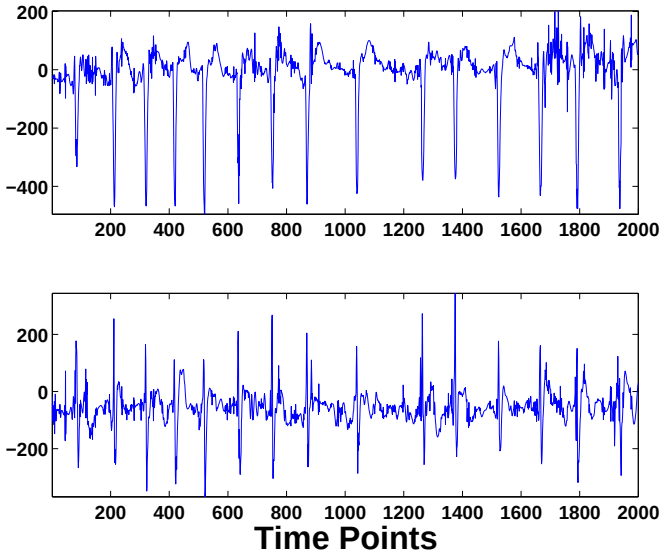


Figure VII.13 Recovered ECG recordings by ISL0 (using the Symmelet-8 wavelet) at CR=70. They cannot be used for diagnosis, since one cannot ensure whether the P waves exist or not.

Chapter VIII

Application: Compressed Sensing of EEG Recordings for Energy-Efficient Wireless Telemonitoring

In the previous two chapters we applied BSBL algorithms and STSBL algorithms to the compressed sensing of ECG recordings with successes. They recovered ECG recordings directly, without sorting to the help of any transformed domains. For example, when BSBL algorithms recovered ECG recordings, they adopted the following model:

$$\mathbf{y} = \mathbf{\Phi x}, \tag{VIII.1}$$

where $\mathbf{y}$ is the compressed data, $\mathbf{\Phi}$ is the sensing matrix, and $\mathbf{x}$ is the original ECG recording. The successes of BSBL algorithms mainly due to the ability to exploit the correlation structure in ECG signals.

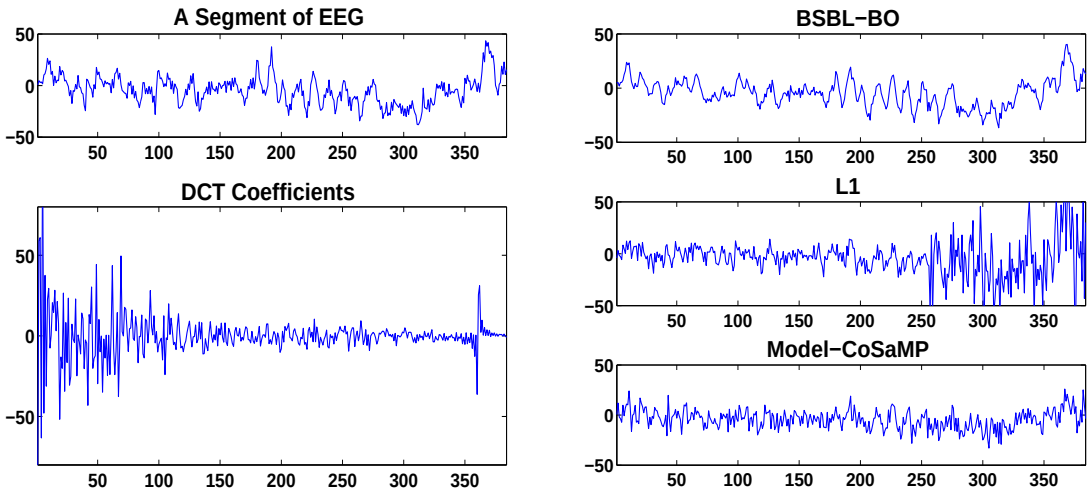
For EEG signals, exploiting the correlation structure in the time domain does not bring large benefit, probably due to serious contamination of noise (for EEG signals, the SNR is generally below 0 dB). So we consider the following model (for BSBL):

$$\mathbf{y} = \mathbf{\Phi D z} \tag{VIII.2}$$

where the original EEG signal $\mathbf{x}$ is represented as $\mathbf{x} = \mathbf{Dz}$, and $\mathbf{D}$ is a orthonormal basis matrix of a transform domain, such as the wavelet domain or the DCT domain. Hopefully, $\mathbf{z}$ can be somewhat sparser than the original signal $\mathbf{x}$, and thus the recovery problem becomes easier than in the model (VIII.1).

The following experiments compared BSBL-BO with some representative CS algorithms in terms of recovery quality. Because all the CS algorithms adopted the same sensing matrix, they had equal energy consumption. Thus, the comparison of energy consumption is excluded.

Two performance indexes were used to measure recovery quality. One was the Normalized Mean Square Error (NMSE), defined as $\|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2 / \|\mathbf{x}\|_2^2$, where $\widehat{\mathbf{x}}$ was the estimate of the true signal $\mathbf{x}$. The second was the Structural SIMilarity index (SSIM) [146] for one-dimensional signals (the length of the sliding window was 100). SSIM measures the similarity between the recovered signal and the



(a) (b)

Figure VIII.1 (a) An EEG epoch, and its DCT coefficients. (b) The recovery results by BSBL-BO, ℓ_1, and Model-CoSaMP when using the model (VIII.2).

original signal, which is a better performance index than the NMSE for structured signals. Higher SSIM means better recovery quality. When the recovered signal is the same as the original signal, SSIM = 1.

In the first experiment $\mathbf{D}$ was the orthonormal basis of the DCT, and thus $\mathbf{z}$ ($\mathbf{z} = \mathbf{D}^{-1}\mathbf{x}$) are DCT coefficients. In the second experiment $\mathbf{D}$ was the orthonormal basis of the Daubechies-20 Wavelet Transform (WT) matrix, which was suggested in [51] for compressing EEG. In both experiments the sensing matrices $\mathbf{\Phi}$ were sparse binary matrices, in which every column contained 15 entries equal to 1 with random locations while other entries were zeros. For BSBL-BO, we defined a block partition, where the starting location of each block was incremented by 24 (i.e., $1, 25, 49, \cdots$). The maximum number of iterations for BSBL-BO was set to seven.

VIII.A Compressed Sensing with DCT

This example used a common dataset ('eeglab_data.set') in the EEGLab [30] to mimic the telemonitoring scenario by first compressing it and then recovering

it. This dataset contains EEG signals of 32 channels with sequence length of 30720 data points, and each channel signal contains 80 epochs each containing 384 points. Artifacts caused by muscle movement are also contained in the signals.

To compress the signals epoch by epoch, we used a 192×384 sparse binary matrix as the sensing matrix $\mathbf{\Phi}$, and a 384×384 inverse DCT matrix as the dictionary matrix $\mathbf{D}$.

Two representative CS algorithms were compared in this experiment. One was the Model-CoSaMP [9], which has high performance for signals with known block structure. Here it used the same block partition as BSBL-BO. The second was an ℓ_1 algorithm used in [51] to recover EEG. The parameters of the two algorithms were tuned for optimal results.

Figure VIII.1(a) shows an EEG epoch and its DCT coefficients. Clearly, the DCT coefficients were not sparse and had no block structure. Figure VIII.1(b) shows the recovery results of the three algorithms. Only BSBL-BO recovered the epoch with good quality; characteristic EEG peaks/troughs and oscillatory activities were accurately presented in the recovered signal. Table VIII.1 shows the averaged NMSE and SSIM of the three algorithms on the whole dataset. It also lists the results when BSBL-BO directly recovered the signals without using the dictionary matrix (i.e., using the model (VIII.1)). The DCT-based BSBL-BO evidently had the best performance, and it took 0.105 second per epoch on average on a computer with 2.8G CPU and 6G RAM. BSBL-BO without using the dictionary matrix took 0.271 second per epoch on average.

In EEG analysis, a regular methodology is performing Independent Component Analysis (ICA) on scalp EEG data and then analyzing single-trial ERPs for each Independent Component (IC) [75]. Therefore, it is important to examine whether the obtained ICs from the recovered EEG dataset by BSBL-BO are the same as those from the original dataset[1].

This study performed ICA decomposition on the original EEG dataset and

[1]We only need to pay attention to the ICs with large energy, since in regular ICA analysis of EEG, ICs with large energy are reliable and meaningful.

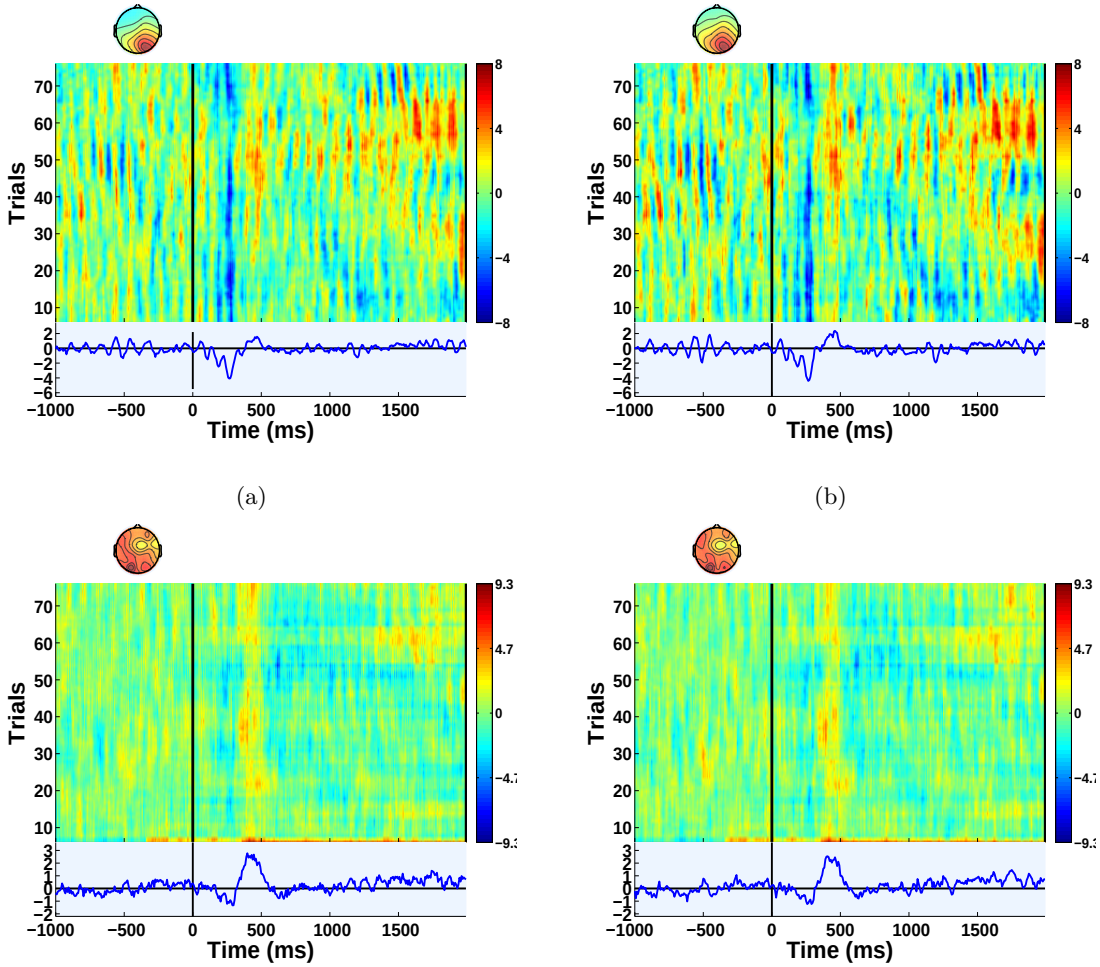


Figure VIII.2 An IC with focal back-projected scalp distribution derived (a) from the original EEG dataset and (b) from the recovered dataset. Another IC with dispersive scalp distribution derived (c) from the original EEG dataset and (d) from the recovered dataset.

Table VIII.1 Averaged NMSE and SSIM of the compared algorithms when they first recovered the DCT coefficients and then recovered the original signals. The results of BSBL-BO when directly recovered the original signals are also given.

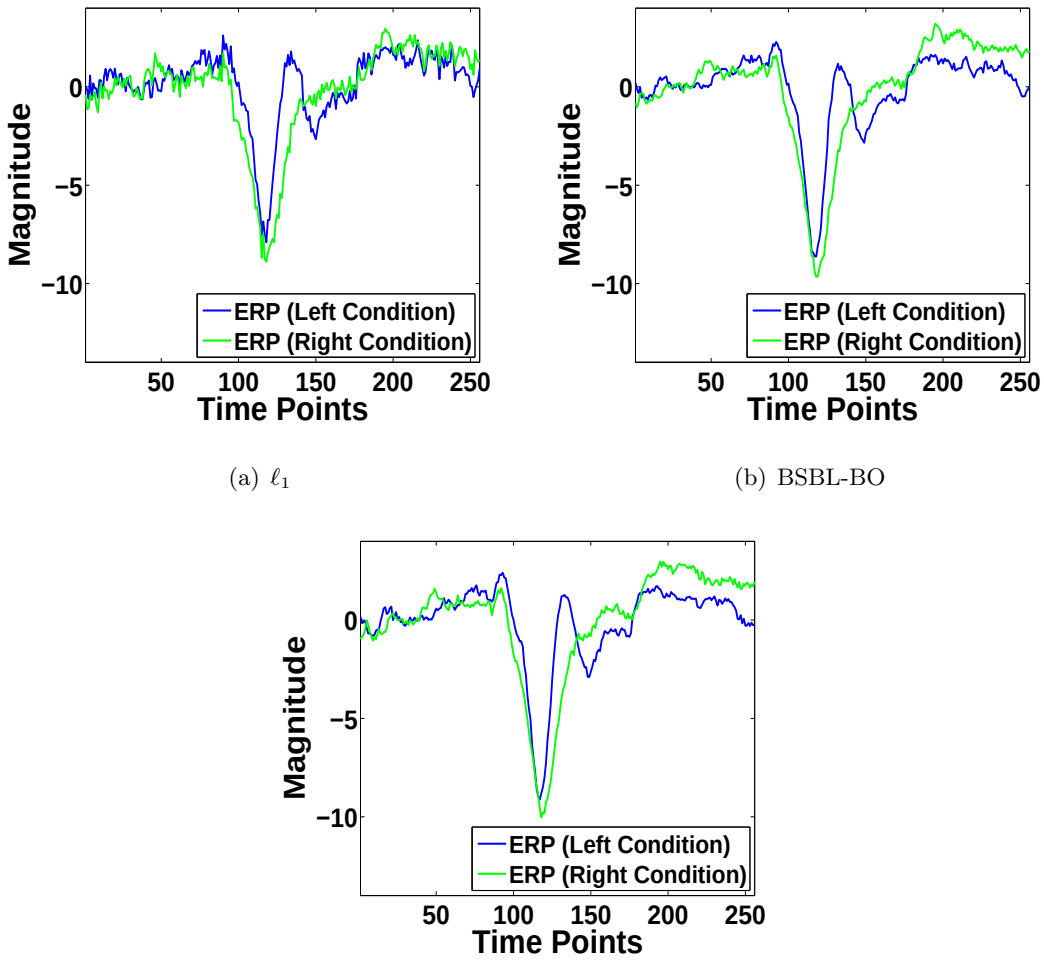
	NMSE (mean ± std)	SSIM (mean ± std)
DCT-based BSBL-BO	**0.078 ± 0.046**	**0.85 ± 0.08**
BSBL-BO without DCT	0.116 ± 0.066	0.81 ± 0.09
DCT-based ℓ_1	0.493 ± 0.121	0.48 ± 0.11
DCT-based Block-CoSaMP	0.434 ± 0.070	0.45 ± 0.10

the recovered EEG dataset by BSBL-BO, respectively, using the Extended-Infomax algorithm with the same initialization, which is a build-in program in the EEGLab [30]. Then, we calculated the back-projected scalp map, the ERP image [75], and the averaged ERP of each IC from the original dataset and the reconstructed dataset.

Figure VIII.2 shows the results of two typical ICs (with large energy) from the recovered dataset (Figure VIII.2 (b)(d)), and the results of corresponding ICs from the original dataset (Figure VIII.2 (a)(c)). Each subfigure shows the back-projected scalp map, the ERP image, and the averaged ERP of an IC. Comparing Figure VIII.2 (a) with (b) and Figure VIII.2 (c) with (d) reveals that there is little difference in terms of scalp maps, ERP images, and averaged ERPs. This implies that BSBL-BO can recover EEG signals with satisfactory quality, ensuring subsequent signal analysis with high fidelity.

VIII.B Compressed Sensing with WT

The second experiment used the dataset in [145]. It consists of multiple channel signals, each channel signal containing 250 epochs for each of two events ('left direction' and 'right direction'). Each epoch consists of 256 sampling points. The goal in [145] is to differentiate the averaged ERP for the 'left direction' with



(a) ℓ_1

(b) BSBL-BO

(c) Original Dataset

Figure VIII.3 The ERPs corresponding to two event conditions ('left' and 'right') averaged (a) from the recovered epochs by the ℓ_1 algorithm, (b) from the recovered epochs by BSBL-BO, and (c) from the original dataset.

the averaged ERP for the 'right direction'. For simplicity, we randomly chose a channel signal from the left parietal area. BSBL-BO and the previous ℓ_1 algorithm were compared. The sensing matrix $\mathbf{\Phi}$ had the size of 128×256, and the dictionary matrix $\mathbf{D}$ had the size of 256×256.

For each event, we calculated the ERP by averaging the associated 250 recovered epochs. Figure VIII.3 (a) shows the ERP for the 'left direction' and the ERP for the 'right direction' averaged from the dataset recovered by the ℓ_1 algorithm. Figure VIII.3 (b) shows the two ERPs averaged from the recovered dataset by BSBL-BO. Figure VIII.3 (c) shows the averaged ERPs from the original dataset (called genuine ERPs). Clearly, the resulting ERPs by the ℓ_1 algorithm were noisy. Although they maintained the main peaks of both genuine ERPs, they did not maintain other details of the genuine ERPs. Particularly, the difference between the two resulting ERPs from the 160^{th} to the 250^{th} time points was not clear. Besides, we found there were many brief oscillatory bursts in the recovered epochs by the ℓ_1 algorithm (due to space limit we omit the results here). In contrast, the ERPs averaged from the recovered epochs by BSBL-BO maintained all the details of the genuine ERPs with high fidelity.

The SSIM and the NMSE of the resulting ERPs by the ℓ_1 algorithm were 0.92 and 0.044, respectively. In contrast, the SSIM and the NMSE of the resulting ERPs by BSBL-BO were 0.97 and 0.008, respectively. In the experiment BSBL-BO took 0.06 second per epoch on average on the previous computer.

VIII.C Conclusion

Compressing EEG for telemonitoring is extremely difficult for current CS algorithms, because EEG is not sparse in the time domain nor sparse in transformed domains. To alleviate the problem, we adopt the BSBL framework, which has superior performance to other existing CS algorithms in recovering non-sparse signals. Experimental results showed that it recovered EEG signals with satisfac-

tory quality, ensuring subsequent signal analysis had high fidelity.

VIII.D Acknowledgements

The text of Chapter VIII, in full, is based on the material as it appears in: Zhilin Zhang, Tzyy-Ping Jung, Scott Makeig, Bhaskar D. Rao, "Compressed Sensing of EEG for Wireless Telemonitoring with Low Energy Consumption and Inexpensive Hardware", to appear in IEEE Trans. on Biomedical Engineering, 2013. The dissertation author was a primary researcher and author of the cited paper.

Chapter IX

Application: Feature Selection for Predicting Patients' Cognitive Levels from Their Neuroimaging Measures

Alzhiemer's disease (AD) is a neurodegenerative disorder characterized by progressive impairment of memory and other cognitive functions. Substantial attention has recently been given to identifying neuroimaging predictors for cognitive decline in AD in the fields of medical image analysis and pattern recognition. Regression models have been investigated to predict patients' cognitive levels from individual magnetic resonance imaging (MRI) and/or positron emission tomography (PET) scans [125, 142, 144, 164]. In [142], stepwise regression was performed in a univariate, pairwise fashion to relate each imaging measure to each cognitive score. In [125], using relevance vector regression, morphometric features of the entire brain were jointly analyzed to predict each selected cognitive score. Two most recent studies [144, 164] employed multi-task learning strategies and aimed to select features that could predict all or most cognitive scores, using $\ell_{2,1}$-norm coupled with ℓ_1-norm [144] and multi-task feature selection coupled with support vector machine [164]. Both methods used a simple concatenation to bundle multiple cognitive scores together without learning their dependence relation.

In this chapter we apply T-MSBL and ST-SBL to this application. The used algorithms are evaluated in empirical studies using the MRI and cognitive data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [147]. These SBL algorithms not only demonstrate superior performance over a number of state-of-the-art competing methods, but also identify cognition-relevant imaging biomarkers that are consistent with prior knowledge.

IX.A Problem Statement and Model Description

The goal of this practical problem is to predict subjects' cognitive scores in a number of neuropsychological assessments using their MRI measures across the entire brain. Each assessment typically yields multiple evaluation scores from a set of relevant cognitive tasks, and thus these scores are inherently correlated. It is hypothesized that only a subset of brain regions are relevant to each assessment.

To achieve the goal, there are two steps. First, using the training dataset, a multivariate regression model is adopted to connect the cognitive scores of all subjects to their MRI measures, and estimate the regression coefficient matrix. The significantly nonzero entries in the coefficient matrix indicate relevant brain regions (or imaging biomarkers). The second step is to predict the cognitive scores of a new subject (in the testing dataset) using his/her MRI measures and the estimated regression coefficient matrix, and evaluate the accuracy.

The multivariate regression model is expressed as follows:

$$\begin{aligned} \mathbf{Y} &= \mathbf{\Phi X} + \mathbf{V} \\ &= \begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} & \cdots & \Phi_{1,N} \\ \Phi_{2,1} & \Phi_{2,2} & \cdots & \Phi_{2,N} \\ \vdots & & \ddots & \\ \Phi_{M,1} & \Phi_{M,2} & \cdots & \Phi_{M,N} \end{bmatrix} \mathbf{X} + \mathbf{V} \end{aligned} \tag{IX.1}$$

where $\mathbf{Y} \triangleq [\mathbf{Y}_{\cdot 1}, \cdots, \mathbf{Y}_{\cdot L}] \in \mathbb{R}^{M \times L}$, $\mathbf{X} \triangleq [\mathbf{X}_{\cdot 1}, \cdots, \mathbf{X}_{\cdot L}] \in \mathbb{R}^{N \times L}$, and $\mathbf{V} \triangleq [\mathbf{V}_{\cdot 1}, \cdots, \mathbf{V}_{\cdot L}]$. Here $\mathbf{Y}_{\cdot l} \in \mathbb{R}^{M \times 1}$ is the cognitive scores of all the M subjects when performing the l-th cognitive task. $\Phi_{j,k}$ is the MRI measure of the k-th brain area of the j-th subject. $\mathbf{X}_{\cdot l}$ is the regression coefficient vector under the l-th task. A significantly nonzero entry of $\mathbf{X}_{\cdot l}$, say $X_{q,l}$, means that the MRI measures of the q-th brain area have strong influence on the cognitive scores of all subjects under the l-th task.

In this model (IX.1), there are two specific structures in $\mathbf{X}$ based on some basic neuroscience observations.

One is **the row-sparse structure**. Since the multiple tasks have inherent connections, when a subject performs a certain cognitive task, if a brain region is relevant, its corresponding MRI measure not only has impact on the cognitive score under this task, but also has more or less influence on the cognitive scores under other tasks. This can be better understood from the expression for the i-th subject:

$$\mathbf{Y}_{i\cdot} = \Phi_{i,1}\mathbf{X}_{1\cdot} + \Phi_{i,2}\mathbf{X}_{2\cdot} + \cdots + \Phi_{i,N}\mathbf{X}_{N\cdot} + \mathbf{V}_{i\cdot}.$$

Further, since all the tasks are relevant to only a few common brain regions, $\mathbf{X}$ has only a few nonzero rows. These row-sparse structure has been exploited in a number of published studies.

The second is **the correlation among entries of the same row in X**. From the above observation, the entries in the same nonzero row of $\mathbf{X}$ do not necessarily have the same value, but their values are highly correlated.

After the MAP estimate of $\mathbf{X}$, denoted by $\widehat{\mathbf{X}}_{\text{MAP}}$, is obtained, the cognitive scores of a new subject under the same cognitive tasks can be predicted by $\boldsymbol{\varphi}\widehat{\mathbf{X}}_{\text{MAP}}$, where $\boldsymbol{\varphi}$ is a row vector consisting of the subject's MRI measures of all the brain areas.

IX.B Use of T-MSBL: Exploiting Correlation Within Coefficient Rows

In this section we use T-MSBL and T-MSBL-FP to solve this problem. As we have seen in Chapter III, the two algorithms can exploit correlation within coefficient rows for better performance.

IX.B.1 Datasets

Data used in the preparation of this article were obtained from the ADNI database (adni.loni.ucla.edu). One goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD. All the healthy control (HC) and AD participants with no missing cognitive and MRI measures were included in this study. Their characteristics are summarized in Table IX.1.

For one baseline scan of each participant, FreeSurfer V4 was employed to automatically label cortical and subcortical tissue classes [28, 50] and to extract target region volume and cortical thickness, as well as to extract total intracranial

Table IX.1 Participant characteristics including gender, handedness, age, and education.

Category	HC	AD	p-value
Gender (M/F)	114/108	86/85	0.835
Handedness (R/L)	205/17	161/10	0.482
Baseline Age (years)	75.93 ± 5.08	75.67 ± 7.36	0.680
Education (years)	15.97 ± 2.84	14.74 ± 3.08	< 0.001

Table IX.2 Description of MMSE, RAVLT ('TOTAL', 'T30', and 'RECOG'), and TRAILS ('TRAILSA', 'TRAILSB' and 'TR(B-A)').

Score	Description
MMSE	MMSE total score
TOTAL	Total score of the first 5 different trials
T30	30 minute delay total number of words recalled
RECOG	30 minute delay recognition score
TRAILSA	Trail making A score
TRAILSB	Trail making B score
TR(B-A)	TRAILSB-TRAILSA

volume (ICV). For each hemisphere, thickness measures of 34 cortical regions of interest (ROIs) and volume measures of 15 cortical and subcortical ROIs were included in this study. Three sets of baseline cognitive scores [4] were employed to test the proposed methods: Mini-Mental State Exam (MMSE), Rey Auditory Verbal Learning Test (RAVLT), and Trail Making (TRAILS). RAVLT includes three dependent scores: 'TOTAL', 'T30', and 'RECOG'. And TRAILS also includes three dependent scores: 'TRAILSA', 'TRAILSB' and 'TR(B-A)'. Details about these scores are available in the ADNI procedure manuals (www.adni-info.org). Table IX.2 summarizes these cognitive scores. Using the regression coefficients derived from the healthy participants, all the FreeSurfer measures were adjusted for the baseline age, gender, education, handedness, and ICV, and all the cognitive measures were adjusted for the baseline age, gender, education and handedness.

Table IX.3 Comparison of cross-validation prediction performances measured by correlation coefficients

Score	T-MSBL-FP	T-MSBL	MFOCUSS	Mixed ℓ_2/ℓ_1	SOMP	RIDGE	MT-CS
MMSE	**0.735**	**0.735**	0.690	0.689	0.721	0.685	0.680
TOTAL	**0.634**	0.617	0.589	0.586	0.604	0.570	0.579
T30	**0.586**	0.572	0.550	0.543	0.545	0.486	0.512
RECOG	**0.561**	0.559	0.526	0.501	0.539	0.504	0.509
TRAILA	**0.467**	0.450	0.391	0.380	0.400	0.312	0.344
TRAILB	**0.565**	0.555	0.491	0.461	0.508	0.464	0.476
TR(B-A)	**0.488**	0.464	0.401	0.351	0.409	0.336	0.355

IX.B.2 Algorithms in the Comparison

In this experiment we used both the T-MSBL algorithm and the T-MSBL-FP algorithm. To show their superior performance, we also selected several state-of-the-art or classical algorithms for comparison. Each algorithm represents a group of methods using different frameworks. They are the Mixed ℓ_2/ℓ_1 Program [44], MFOCUSS [25], Simultaneous Orthogonal Matching Pursuit (SOMP) [136], Multi-Task Compressive Sensing (MT-CS) [71], and Ridge Regression. Among these algorithm, MT-CS treats the model (IX.1) as L *dependent* single measurement vector (SMV) models, i.e., $\mathbf{Y}_{\cdot i} = \mathbf{\Phi}\mathbf{X}_{\cdot i} + \mathbf{V}_{\cdot i}$ $(i = 1, \cdots, L)$, where every $\mathbf{X}_{\cdot i}(\forall i)$ shares a common prior. This model is an alternative one to the MMV model in multi-task learning. Ridge Regression is a traditional regression approach for an SMV model. To use it in our problem, we applied it to each $\mathbf{Y}_{\cdot i} = \mathbf{\Phi}\mathbf{X}_{\cdot i} + \mathbf{V}_{\cdot i}$ $(i = 1, \cdots, L)$ independently.

IX.B.3 Results of Prediction

Regression was performed separately on each cognitive task (MMSE, RAVLT, or TRAILS) using the MRI measures as predictors, where the proposed T-MSBL-FP method and all the competing methods (T-MSBL, MFOCUSS, Mixed ℓ_2/ℓ_1, SOMP, RIDGE, MT-CS) were evaluated. Similar to prior studies [125, 164], in each experiment, Pearson's correlation coefficients r between the actual and predicted

cognitive scores were computed to measure the prediction performance. Using a 5-fold cross-validation strategy, the testing samples across five trials were pulled together to obtain an unbiased estimate of these correlation coefficients.

Shown in Table IX.3 is the performance comparison among all seven methods. Both T-MSBL-FP and T-MSBL outperformed the other five competing algorithms in all three prediction cases.

In particular, using T-MSBL-FP, the MRI measures could predict the MMSE score the best, with a correlation coefficient $r = 0.7352$. This result is better than or competitive to a few prior prediction results on the MMSE score: $r = 0.504$ using MRI only in [164], $r = 0.697$ using MRI, PET and CSF jointly in [164], and $r = 0.70$ using MRI in [125]. Relatively high prediction performance has also been achieved for RAVLT scores (i.e., the TOTAL, T30, and RECOG scores), from $r = 0.561$ to $r = 0.634$. In [125], a different, but relevant RAVLT score was predicted using MRI, with $r = 0.13$ only.

IX.B.4 Results of Biomarker Identification

Both T-MSBL-FP and T-MSBL are based on the sparse model that are able to identify a compact set of relevant neuroimaging biomarkers and to explain the underlying brain structural changes related to cognitive status. Shown in Figure IX.1 are the heat maps of the regression weights (or coefficients) of the MRI measures for each cognitive score calculated by T-MSBL-FP, T-MSBL, and the Mixed ℓ_2/ℓ_1 Program (for illustration only the heatmap from the Mixed ℓ_2/ℓ_1 Program is shown here). In the picture, results for volume measures are shown in top 15 rows, and those for thickness measures are shown in bottom 34 rows; results for left (L) and right (R) hemispheres are shown in separate panels. Blue indicates negative correlation, while red indicates positive correlation. The bigger the magnitude of an coefficient is, the more important its MRI measure is in predicting the corresponding cognitive score.

T-MSBL-FP and T-MSBL clearly yielded a much sparser pattern than the

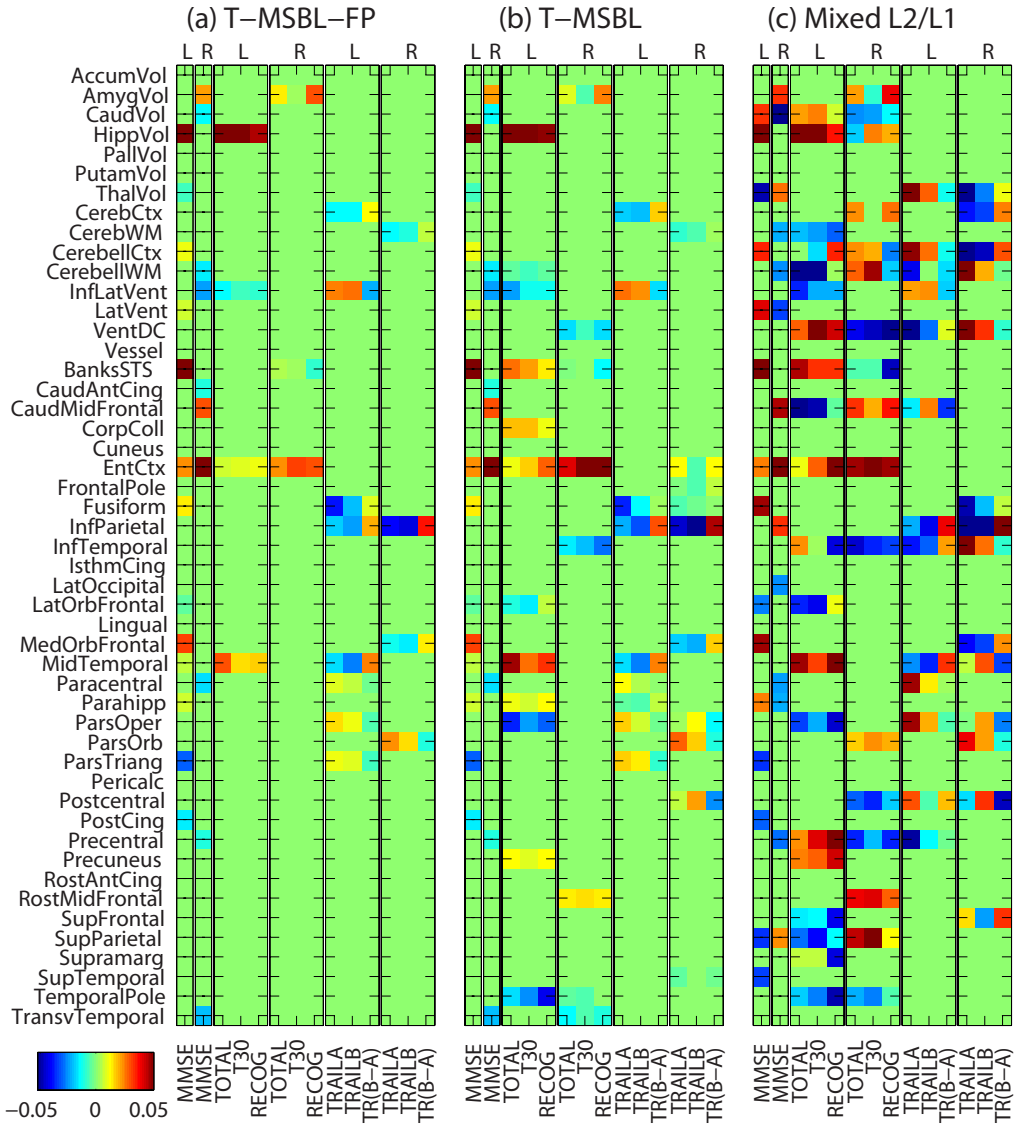


Figure IX.1 Heat maps of average regression coefficients of 5-fold cross-validation trials for (a) T-MSBL-FP, (b) T-MSBL, and (c) Mixed ℓ_2/ℓ_1. Each row corresponds to an MRI measure and each column to a cognitive score.

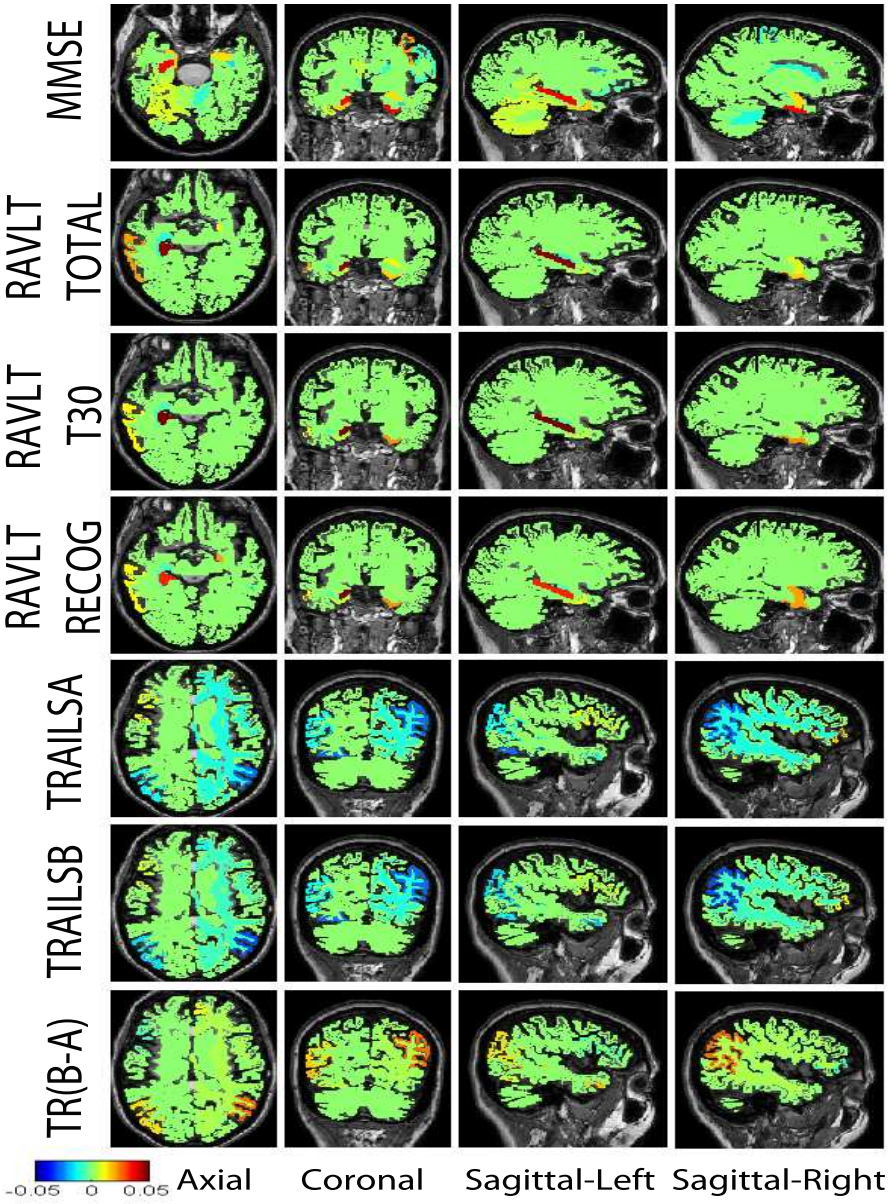


Figure IX.2 Regression coefficients mapped onto brain: Each row corresponds to one cognitive score. Each column corresponds to a specific view of the brain.

Mixed ℓ_2/ℓ_1 (Figure IX.1), making the results easier to interpret. The patterns obtained by T-MSBL-FP and T-MSBL were also much sparser and cleaner than those obtained by other algorithms (not shown here).

Figure IX.2 shows these regression coefficients mapped on the brain, where

each row corresponds to one cognitive score and each column corresponds to a specific view of the brain.

The imaging biomarkers identified by T-MSBL-FP yielded promising patterns (Figure IX.2) that are expected based on prior knowledge on neuroimaging and cognition. MMSE measures overall cognitive impairment; and thus its result includes important AD-relevant imaging markers such as hippocampal volume, amygdala volume, and entorhinal cortex thickness. RAVLT measures verbal learning memory; and thus its result includes regions relevant to learning and memory, such as hippocampus, entorhinal cortex, and middle temporal gyri. TRAILS measures a combination of visual, motor and executive functions; and thus its result includes regions in sensory-motor cortex (e.g., paracentral lobule), parietal lobe (relevant to visual processing), and frontal lobe (relevant to executive function).

All the above results have demonstrated that the proposed T-MSBL-FP method not only yields superior performance on prediction accuracy and computational time, but also is a powerful tool for discovering a small set of imaging biomarkers that predict cognitive performance. These results provide important information for understanding brain structural changes related to cognitive status and can potentially help characterize the progression of AD.

IX.C Use of STSBL: Exploiting both Correlation and Nonlinear Relationship

In the previous section we use T-MSBL and T-MSBL-FP to solve this problem. The model they are based on is an MMV model. However, one possible limitation in this model is that a subject's cognitive score under a task is modeled as a **linear** function of his/her MRI measures. For example, for the m-th subject, the cognitive score under the l-th task is modeled as

$$Y_{m,l} = \Phi_{m,1} X_{1,l} + \Phi_{m,2} X_{2,l} + \cdots + \Phi_{m,N} X_{N,l} + V_{m,l}.$$

A linear model might have limited flexibility in capturing the complex relationship between $Y_{m,l}$ and $\Phi_{m,n}$.

A more powerful model is to consider a nonlinear relationship between $Y_{m,l}$ and $\Phi_{m,n}$. To achieve this, we use polynomials to model the nonlinear relationship as follows:

$$\begin{aligned} Y_{m,l} &= \Phi_{m,1}Z_{1,l} + \Phi_{m,1}^2 Z_{2,l} + \cdots + \Phi_{m,1}^{d_1} Z_{d_1,l} \\ &\quad + \cdots + \Phi_{m,N} Z_{c+1,l} + \Phi_{m,N}^2 Z_{c+2,l} \\ &\quad + \cdots + \Phi_{m,N}^{d_g} Z_{c+d_g,l} + V_{m,l} \end{aligned} \tag{IX.2}$$

where $c = \sum_{i=1}^{g-1} d_i$. Note that if the MRI measure $\Phi_{m,1}$ has influence on the subject's cognitive scores, the associated coefficients $Z_{1,l}, Z_{2,l}, \cdots, Z_{d_1,l}$ tend to be nonzero together (but with different amplitudes), and thus they are correlated. This correlation is in fact the intra-block correlation stated before. The same holds for other MRI measures $\Phi_{m,j} (j = 2, \cdots, N)$.

Writing the relation (IX.2) for all m, l, we obtain the following model in matrix form:

$$\mathbf{Y} = \begin{bmatrix} \Phi_{1,1} & \cdots & \Phi_{1,1}^{d_1} & \cdots & \Phi_{1,N}^{d_g-1} & \Phi_{1,N}^{d_g} \\ \Phi_{2,1} & \cdots & \Phi_{2,1}^{d_1} & \cdots & \Phi_{2,N}^{d_g-1} & \Phi_{2,N}^{d_g} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \Phi_{M,1} & \cdots & \Phi_{M,1}^{d_1} & \cdots & \Phi_{M,N}^{d_g-1} & \Phi_{M,N}^{d_g} \end{bmatrix} \mathbf{Z} + \mathbf{V} \tag{IX.3}$$

where the (i, j)-th entry of $\mathbf{Z}$ is $Z_{i,j}$. Note that $\mathbf{Z}$ has the block structure as in (IV.2), and has the intra-block correlation. $\mathbf{Z}$ also has the correlation within coefficient rows (which is inherited from the model (IX.1)). Therefore, the model (IX.3) is exactly the spatiotemporal sparse model (IV.1), and thus both STSBL-BO and STSBL-EM can be directly used.

In the model (IX.3) one can use other nonlinear functions instead of the polynomials. A future direction is to explore various choices of nonlinear functions. It is worth noting that although the nonlinear relationship between cognitive scores

Table IX.4 Comparison of prediction performance measured by mean of the correlation coefficients.

Score	STSBL-BO	T-MSBL-FP	MFOCUSS	Mixed ℓ_2/ℓ_1	SOMP	RIDGE	MT-CS
ADAS	**0.767**	0.753	0.749	0.740	0.760	0.746	0.746
MMSE	**0.758**	0.740	0.733	0.718	0.725	0.731	0.726
TOTAL	**0.633**	0.617	0.597	0.608	0.612	0.606	0.611
T30	**0.608**	0.586	0.571	0.569	0.575	0.545	0.556
RECOG	**0.598**	0.567	0.545	0.549	0.543	0.551	0.544
TRAILA	**0.562**	0.504	0.487	0.488	0.502	0.476	0.477
TRAILB	**0.607**	0.573	0.575	0.550	0.541	0.537	0.542
TR(B-A)	**0.525**	0.502	0.512	0.472	0.457	0.427	0.439

and MRI measures has been studied in other models, it is the first time that this relationship is exploited in sparse models.

In the following experiment, we used the third-order polynomial in (IX.3), i.e., $d_1 = d_2 = \cdots = d_g = 3$. The dataset was almost the same as in the previous section, and we also added another set of cognitive scores and associated MRI measures, namely the Alzheimer's Disease Assessment Scale (ADAS). Algorithms compared were also the same.

IX.C.1 Results of Prediction

All the algorithms were applied to predicting the cognitive scores using the MRI measures. For each set of scores (i.e., ADAS, MMSE, RAVLT, and TRAILS), five-fold cross-validation was used to obtain the mean of r. The results (Table IX.4) showed that STSBL-BO outperformed all the compared algorithms for all four sets of scores. Furthermore, comparing STSBL-BO to T-MSBL-FP, we see the prediction accuracy was improved significantly. Since the linear model used by T-MSBL-FP is a special case of the nonlinear model used by STSBL-BO when $d_1 = \cdots = d_g = 1$, we can see the nonlinear model, with the ability to exploit the intra-block correlation, can better capture the relation between the predictors (the MRI measures) and the responses (the cognitive scores) than the linear model.

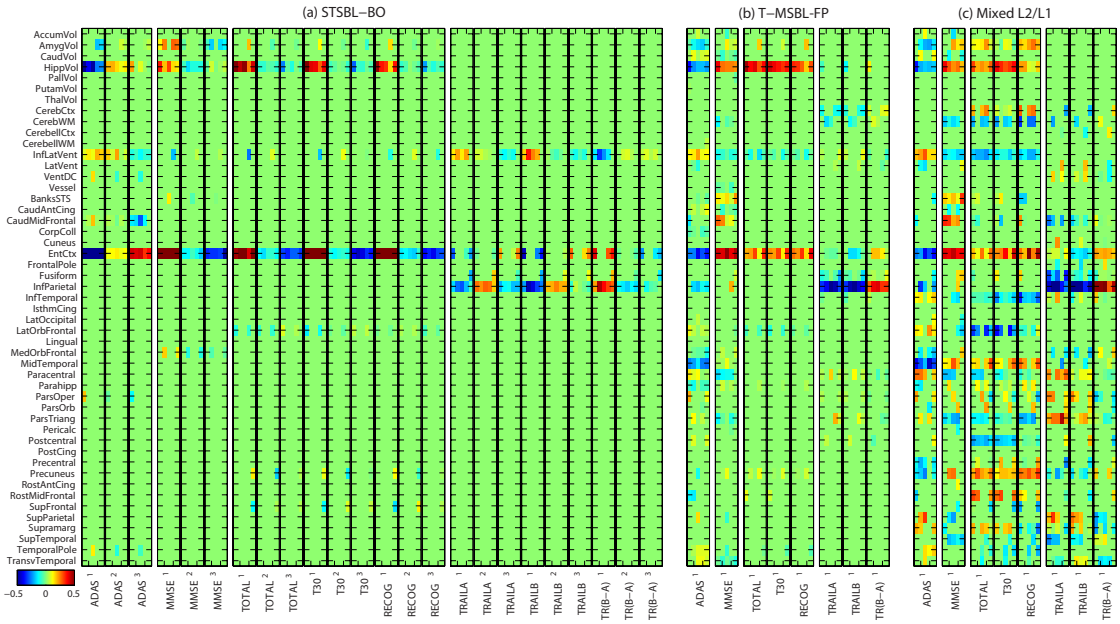


Figure IX.3 Heat maps of regression coefficients of the 5-fold cross-validation trials for (a) STSBL-BO, (b) T-MSBL-FP, and (c) the mixed ℓ_2/ℓ_1 minimization algorithm. Results for volume measures are shown in top 15 rows, and those for thickness measures in bottom 34 rows.

IX.C.2 Results of Biomarker Identification

Figure IX.3 shows the heat maps of the regression coefficients calculated by STSBL-BO, T-MSBL-FP, and the mixed ℓ_2/ℓ_1 minimization algorithm for all the cognitive tasks in the four sets. In the heat maps, every five columns form a column block to represent the regression coefficients in the five-fold cross validation. In Figure IX.3 (b) and (c), each row corresponds to an MRI measure, and each column block correspond to a cognitive task. In Figure IX.3 (a), since we adopted the nonlinear model using the third-order polynomial, every *three column blocks* correspond to a cognitive task, representing the regression coefficients corresponding to the 1st, the 2nd, and the 3rd order of the polynomial, respectively. In addition, the blue color indicates negative correlation while the red indicates positive correlation and the green indicates zero correlation. The larger the value of the coefficient, the more important its corresponding MRI measure is in predicting

the corresponding cognitive score.

The pattern obtained by STSBL-BO is more sparse than those obtained by T-MSBL-FP, Mixed ℓ_2/ℓ_1 and other compared algorithms (not shown due to space constraint), making the results easier to interpret. The imaging biomarkers identified by STSBL-BO yielded promising patterns (Figure IX.3) that are expected from prior knowledge on neuroimaging and cognition. The tasks in ADAS and MMSE aimed to reflect overall cognitive impairment, while the result of STSBL-BO showed important AD-relevant imaging biomarkers such as hippocampal volume (HippVol), amygdala volume (AmygVol), and entorhinal cortex thickness (EntCtx). The tasks in RAVLT (i.e., TOTAL, T30 and RECOG) aimed to test verbal learning memory, while the result of STSBL-BO highlighted regions relevant to learning and memory, such as hippocampus (HippVol) and entorhinal cortex (EntCtx). The tasks in TRAILS (i.e., TRAILSA, TRAILSB, TR(B-A)) aimed to test a combination of visual, motor and executive functions, while the result of STSBL-BO showed regions in temporal lobe (EntCtx, Fusiform), parietal lobe (InfParietal), and ventricle (InfLatVent).

All the above results demonstrated that the proposed STSBL-BO algorithm not only yields higher prediction accuracy, but also has a desired ability to discover a small set of imaging biomarkers that are easier to interpret and are consistent with prior neuroscience knowledge. The algorithm can provide important information for understanding brain structural changes related to cognitive status, and can potentially help characterize the progression of AD.

IX.D Conclusion

In this chapter, we applied three algorithms to predict cognitive scores of subjects from their MRI measures. They were T-MSBL, T-MSBL-FP, and STSBL-BO. Since the first two algorithms are based on the canonical multiple measurement vector model, they only exploited correlation within each row of the

solution matrix. The third algorithm is based on the spatiotemporal sparse model studied in Chapter IV. Thus, it not only exploited correlation within each row of the solution matrix, but also exploited intra-block correlation in each column of the solution matrix. Besides, due to the construction of the predictor matrix, it also exploited the nonlinear relationship between response variables and predictor variables. Compared to state-of-the-art algorithms, the three algorithm not only showed the highest prediction accuracy, but also demonstrated the ability to accurately identify imaging biomarkers that are consistent with prior knowledge. It is worth highlighting that the used algorithms can be applied to many other domains, such as variable selection in high dimensional space, compressed sensing, and spatiotemporal data mining.

IX.E Acknowledgements

The text of Chapter IX, in full, is based on the material as it appears in: Zhilin Zhang, Jing Wan, Shiaofen Fang, Andrew Saykin, Li Shen, "Correlation- and Nonlinearity-Aware Sparse Bayesian Learning with Applications to the Prediction of Cognitive Scores from Neuroimaging Measures", submitted to IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013, and Jing Wan, Zhilin Zhang, Jingwen Yan, Taiyong Li, Bhaskar D. Rao, Shiaofen Fang, Sungeun Kim, Shannon Risacher, Andrew Saykin, Li Shen, "Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer's Disease", in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2012. The dissertation author was a primary researcher and author of the cited papers.

Chapter X

Conclusions

A trend in sparse signal recovery is to exploit more information in addition to sparsity in signals to achieve better performance. In many applications, signals are not independent and identically distributed, but have rich structures. Recently developed algorithms have considered a variety of structures such as group structures, tree structures, low-rank structures and so on. However, few works seriously considered correlation among amplitudes of signals. Thus, it is unclear what role the correlation plays in signal recovery.

In this dissertation, we first proposed a block sparse Bayesian learning (BSBL) framework. Based on this framework, we derived a number of algorithms which exploit intra-block correlation in a canonical block sparse model, temporal correlation in a canonical multiple measurement vector model, spatiotemporal correlation in a spatiotemporal sparse model, and local temporal correlation in a time-varying sparse model. Further, by connecting these algorithms to popular algorithms including Group-Lasso type algorithms and iterative reweighted ℓ_1 and ℓ_2 algorithms, we suggested a procedure for modifying these algorithms such that they can also exploit the correlation structure for better performance. These algorithms largely enrich the family of sparse signal recovery algorithms. More importantly, these algorithms demonstrate effective ways (using the Bayesian framework or using the iterative reweighted framework) to exploit the correlation structure, and motivate more studies on this topic.

The benefit of exploiting the correlation structure is not only shown through computer simulations, but also shown in several challenging practical problems. One is compressed sensing of raw physiological signals for energy-efficient wireless telemonitoring. In this application, signals are not sparse and also not sufficiently sparse in any transformed domain. Consequently, existing compressed sensing algorithms are unable to achieve satisfactory results. But using the derived algorithms, we have achieved satisfactory results for clinical diagnosis. Another practical problem is identification of neuroimaging markers for prediction of patients' cognition levels, which is a challenging feature selection problem in medical image

analysis. For this problem, we have achieved much higher prediction accuracy than reported results on some common datasets, and the identified neuroimaging markers are consistent with prior knowledge in neuroscience.

However, our work indicates there are some trade-offs involved while trying to exploit the correlation. When exploiting correlation, one has to address the conflict between limited data and accurate modeling of correlation matrices. Learning correlation matrices means significantly increased number of unknown parameters, which potentially increases the difficulty in recovery of sparse signals. Therefore, regularization is required. This issue is more important in highly noisy environments. Poorly selected regularization strategies could result in worse recovery performance than methods that do not exploit correlation.

Bibliography

[1] *Special Issue on Cyber-Physical Systems, Proceedings of the IEEE*, 2012, vol. 100, no. 1.

[2] P. Addison, "Wavelet transforms and the ECG: a review," *Physiological measurement*, vol. 26, p. R155, 2005.

[3] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio inpainting," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 922–932, 2012.

[4] P. S. Aisen, R. C. Petersen *et al.*, "Clinical core of the alzheimer's disease neuroimaging initiative: progress and plans," *Alzheimers Dement*, vol. 6, no. 3, pp. 239–46, 2010.

[5] D. Angelosante, G. Giannakis, and E. Grossi, "Compressed sensing of time-varying signals," in *Digital Signal Processing, 2009 16th International Conference on*, 2009, pp. 1–8.

[6] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm," *Signal Processing, IEEE Transactions on*, vol. 58, no. 7, pp. 3436–3447, 2010.

[7] S. Babacan, R. Molina, and A. Katsaggelos, "Bayesian compressive sensing using laplace priors," *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 53–63, 2010.

[8] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[9] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[10] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *Information Theory, IEEE Transactions on*, vol. 57, no. 2, pp. 764–785, 2011.

[11] S. Becker, J. Bobin, and E. J. Candes, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[12] J. Bobin, J. Starck, and R. Ottensamer, "Compressed sensing in astronomy," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 5, pp. 718–726, 2008.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] P. Brown, M. Vannucci, and T. Fearn, "Multivariate bayesian variable selection and prediction," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 60, no. 3, pp. 627–641, 1998.

[15] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[16] E. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[17] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J Fourier Anal Appl*, vol. 14, pp. 877–905, 2008.

[18] H. Cao, V. Leung, C. Chow, and H. Chan, "Enabling technologies for wireless body area networks: A survey and outlook," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009.

[19] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homespresent state and future challenges," *Computer methods and programs in biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

[20] F. Chen, A. Chandrakasan, and V. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[21] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[22] Y. Cho and L. K. Saul, "Sparse decomposition of mixed audio signals by basis pursuit with autoregressive models," in *Proc. of the 34th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, Taipei, pp. 1705–1708.

[23] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.

[24] S. F. Cotter, "Multiple snapshot matching pursuit for direction of arrival (DOA) estimation," in *Proc. of the 15th European Signal Processing Conference (EUSIPCO 2007)*, Poznan, Poland, 2007.

[25] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[26] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, Sep. 2012.

[27] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2230–2249, 2009.

[28] A. Dale, B. Fischl, and M. Sereno, "Cortical surface-based analysis. i. segmentation and surface reconstruction." *Neuroimage*, vol. 9, no. 2, pp. 179–94, 1999.

[29] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Fetal electrocardiogram extraction by blind source subspace separation," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 5, pp. 567–572, 2000.

[30] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004.

[31] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," *TR11-06, Department of Computational and Applied Mathematics, Rice University*, 2011.

[32] D. G. Dension, C. C. Holmes, B. K. Mallick, and A. F. Smith, *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley & Sons, LTD, 2002.

[33] A. M. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[34] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosophical Transactions of the Royal Society A*, vol. 367, no. 1906, pp. 4273–4293, 2009.

[35] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Stanford University Technical Report*, 2004.

[36] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization," *PNAS*, vol. 100, no. 5, pp. 2197–2202, 2003.

[37] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[38] D. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[39] M. Duarte, G. Shen, A. Ortega, R. Baraniuk, M. Duarte, G. Shen, A. Ortega, and R. Baraniuk, "Signal compression in wireless sensor networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 118–135, 2012.

[40] C. Eduardo, P. Octavian Adrian, S. Pedro *et al.*, "Implementation of compressed sensing in telecardiology sensor networks," *International Journal of Telemedicine and Applications*, vol. 2010, 2010.

[41] M. Elad, "Sparse representations are most likely to be the sparsest possible," *EUROSIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–12, 2006.

[42] ——, *Sparse and redundant representations*. Springer Verlag, 2010.

[43] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[44] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[45] Y. C. Eldar and H. Rauhut, "Average case analysis of multichannel sparse recovery using convex relaxation," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 505–519, 2010.

[46] E. Elhamifar and R. Vidal, "Block-sparse recovery via convex optimization," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 4094–4107, 2012.

[47] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[48] J. Fan, J. Lv, and L. Qi, "Sparse high dimensional models in economics," *Annual review of economics*, vol. 3, p. 291, 2011.

[49] M. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1150–1159, 2003.

[50] B. Fischl, M. Sereno, and A. Dale, "Cortical surface-based analysis. ii: Inflation, flattening, and a surface-based coordinate system." *Neuroimage*, vol. 9, no. 2, pp. 195–207, 1999.

[51] D. Gangopadhyay, E. Allstot, A. Dixon, and D. Allstot, "System considerations for the compressive sampling of EEG and ECoG bio-signals," in *BioCAS 2011*, pp. 129–132.

[52] S. Ganguli and H. Sompolinsky, "Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis," *Annual Review of Neuroscience*, vol. 35, pp. 485–508, 2012.

[53] D. Giacobello, M. Christensen, M. Murthi, S. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1644–1657, 2012.

[54] A. Goldberger and et al, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[55] I. Gorodnitsky, J. George, and B. Rao, "Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm," *Electroencephalography and clinical Neurophysiology*, vol. 95, no. 4, pp. 231–251, 1995.

[56] I. Gorodnitsky, B. Rao, and J. George, "Source localization in magnetoencephalography using an iterative weighted minimum norm algorithm," in *Signals, Systems and Computers, 1992. 1992 Conference Record of The Twenty-Sixth Asilomar Conference on*, 1992, pp. 167–171.

[57] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[58] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing," *CAAM Technical Report TR07-07, Rice University*, 2007.

[59] P. Hansen, "Analysis of discrete ill-posed problems by means of the l-curve," *SIAM review*, vol. 34, no. 4, pp. 561–580, 1992.

[60] P. Hansen and D. O'Leary, "The use of the l-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1487–1503, 1993.

[61] M. Hasan, M. Reaz, M. Ibrahimy, M. Hussain, and J. Uddin, "Detection and processing techniques of FECG signal for fetal monitoring," *Biological procedures online*, vol. 11, no. 1, pp. 263–295, 2009.

[62] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 92–101, 2008.

[63] F. Herrmann, M. Friedlander, and O. Yilmaz, "Fighting the curse of dimensionality: compressive sensing in exploration seismology," *Signal Processing Magazine, IEEE*, vol. 29, no. 3, pp. 88–100, 2012.

[64] C. Herzet and A. Drémeau, "Bayesian Pursuit Algorithms." [Online]. Available: http://hal.inria.fr/hal-00673801

[65] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 417–424.

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

[67] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[68] A. Hyvärinen, "Optimal approximation of signal priors," *Neural Computation*, vol. 20, no. 12, pp. 3087–3110, 2008.

[69] G. Isely, C. Hillar, and F. Sommer, "Deciphering subsampled data: adaptive compressive sampling as a principle of brain communication," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 910–918.

[70] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[71] S. Ji, D. Dunson, and L. Carin, "Multi-task compressive sensing," *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 92–106, 2009.

[72] Y. Jin and B. Rao, "Support recovery of sparse signals in the presence of multiple measurement vectors," *arXiv preprint arXiv:1109.1895*, 2011.

[73] Y. Jin and B. D. Rao, "Insights into the stable recovery of sparse solutions in overcomplete representations using network information theory," in *Proc. of the 33th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, Las Vegas, USA, 2008, pp. 3921–3924.

[74] ——, "On the role of the properties of the nonzero entries on sparse signal recovery," in *Proc. of the 44th Asilomar Conference on Signals, Systems, and Computers*, USA, 2010, pp. 753–757.

[75] T.-P. Jung, S. Makeig, M. McKeown, A. Bell, T. Lee, and T. Sejnowski, "Imaging brain dynamics using independent component analysis," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1107–1122, 2001.

[76] J. Kim, O. Lee, and J. Ye, "Compressive music: revisiting the link between compressive sensing and array signal processing," *Information Theory, IEEE Transactions on*, vol. 58, no. 1, pp. 278–301, 2012.

[77] K. Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3613–3641, 2012.

[78] T. Li and Z. Zhang, "Face recognition via block sparse Bayesian learning," *submitted to Neurocomputing*, 2012.

[79] B. Liu, Z. Zhang, H. Fan, Z. Lu, and Q. Fu, "Fast marginalized block SBL algorithm," *submitted to IEEE Signal Processing Letters*, 2012.

[80] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

[81] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Springer, 2005.

[82] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[83] D. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[84] ——, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.

[85] S. Makeig, C. Kothe, T. Mullen, N. Bigdely-Shamlo, Z. Zhang, and K. Kreutz-Delgado, "Evolving signal processing for brain–computer interfaces," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1567–1584, 2012.

[86] D. Malioutov, M. Çetin, and A. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 3010–3022, 2005.

[87] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[88] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[89] ——, "Structured sparsity models for compressively sensed electrocardiogram signals: A comparative study," in *Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE*, 2011, pp. 125–128.

[90] A. Martínez, R. Alcaraz, and J. Rieta, "Study on the p-wave feature time course as early predictors of paroxysmal atrial fibrillation," *Physiological Measurement*, vol. 33, no. 12, p. 1959, 2012.

[91] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *The Annals of Statistics*, vol. 37, no. 1, pp. 246–270, 2009.

[92] C. M. Michel, T. Koenig, D. Brandeis, and et al, *Electrical Neuroimaging*. Cambridge University Press, 2009.

[93] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer communications*, vol. 29, no. 13-14, pp. 2521–2533, 2006.

[94] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed l0 norm," *IEEE Trans. on Signal Processing*, vol. 57, no. 1, pp. 289–301, 2009.

[95] G. Moody, W. Muldrow, and R. Mark, "The MIT-BIH noise stress test database." [Online]. Available: http://www.physionet.org/physiobank/database/nstdb

[96] B. D. Moor, "DaISy: Database for the identification of systems," November 2011. [Online]. Available: http://www.esat.kuleuven.ac.be/sista/daisy

[97] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[98] R. M. Neal, *Bayesian learning for neural networks*. Springer, 1996.

[99] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[100] S. Negahban and M. J. Wainwright, "Simultaneous support recovery in high dimensions: benefits and perils of block ℓ_1/ℓ_∞-regularization," *IEEE Trans. on Information Theory*, vol. 57, no. 6, pp. 3841–3863, 2011.

[101] T. Peleg, Y. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2286–2303, 2012.

[102] L. Polania, R. Carrillo, M. Blanco-Velasco, and K. Barner, "Compressive sensing exploiting wavelet-domain dependencies for ECG compression," in *SPIE Defense, Security, and Sensing*, 2012, pp. 83 650E–83 650E.

[103] ——, "On exploiting interbeat correlation in compressive sensing-based ECG compression," in *SPIE Defense, Security, and Sensing*, 2012, pp. 83 650D–83 650D.

[104] L. Potter, E. Ertin, J. Parker, and M. Cetin, "Sparsity and compressed sensing in radar imaging," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1006–1020, 2010.

[105] K. Qiu and A. Dogandzic, "Variance-component based sparse signal reconstruction and model selection," *IEEE Trans. on Signal Processing*, vol. 58, no. 6, pp. 2935–2952, 2010.

[106] B. D. Rao and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," in *Proc. IEEE Digital Signal Processing Workshop*, Bryce Canyon, UT, 1998.

[107] B. Rao, Z. Zhang, and Y. Jin, "Sparse signal recovery in the presence of intra-vector and inter-vector correlation," in *Signal Processing and Communications (SPCOM), 2012 International Conference on*, 2012, pp. 1–5.

[108] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Trans. on Signal Processing*, vol. 51, no. 3, pp. 760–770, 2003.

[109] M. Salman Asif and J. Romberg, "Dynamic updating for ℓ_1 minimization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 421–434, 2010.

[110] R. Sameni and G. Clifford, "A review of fetal ECG signal processing; issues and promising directions," *The open pacing, electrophysiology & therapy journal*, vol. 3, p. 4, 2010.

[111] R. Sameni, C. Jutten, and M. Shamsollahi, "A deflation procedure for subspace decomposition," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2363–2374, 2010.

[112] R. Sameni, "OSET: The open-source electrophysiological toolbox," January 2012. [Online]. Available: http://www.oset.ir/

[113] P. Schniter, L. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Information Theory and Applications Workshop, 2008*, 2008, pp. 326–333.

[114] D. Sejdinovic, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 2010, pp. 1730–1736.

[115] A. Seneviratne and V. Solo, "On vector l0 penalized multivariate regression," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3613–3616.

[116] B. Shahrasbi, A. Talari, and N. Rahnavard, "Tc-csbp: Compressive sensing for time-correlated data based on belief propagation," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*. IEEE, 2011, pp. 1–6.

[117] R. Shepovalnikov, A. Nemirko, A. Kalinichenko, and V. Abramchenko, "Investigation of time, amplitude, and frequency parameters of a direct fetal ECG signal during labor and delivery," *Pattern Recognition and Image Analysis*, vol. 16, no. 1, pp. 74–76, 2006.

[118] D. Shutin, T. Buchgraber, S. Kulkarni, and H. Poor, "Fast variational sparse bayesian learning with automatic relevance determination for superimposed signals," *Signal Processing, IEEE Transactions on*, vol. 59, no. 12, pp. 6257–6261, 2011.

[119] J. Smith Jr, "Fetal health assessment using prenatal diagnostic techniques," *Current Opinion in Obstetrics and Gynecology*, vol. 20, no. 2, p. 152, 2008.

[120] V. Solo, "A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, 1996, pp. 89–92.

[121] J. Starck and J. Bobin, "Astronomical data analysis and sparsity: from wavelets to compressed sensing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1021–1030, 2010.

[122] C. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[123] P. Stoica and P. Babu, "SPICE and LIKES: Two hyperparameter-free methods for sparse-parameter estimation," *Signal Processing*, 2012.

[124] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3075–3085, 2009.

[125] C. Stonnington, C. Chu, S. Klöppel, C. Jack Jr, J. Ashburner, R. Frackowiak *et al.*, "Predicting clinical scores from magnetic resonance scans in alzheimer's disease," *Neuroimage*, vol. 51, no. 4, pp. 1405–13, 2010.

[126] T. Sun and C.-H. Zhang, "Scaled sparse linear regression," *Biometrika*, vol. 99, no. 4, pp. 879–898, 2012.

[127] X. Tan and J. Li, "Computationally efficient sparse bayesian learning via belief propagation," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, 2010.

[128] G. Tang and A. Nehorai, "Performance analysis for sparse support recovery," *IEEE Trans. on Information Theory*, vol. 56, no. 3, pp. 1383–1399, 2010.

[129] 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.

[130] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. Tibshirani, "Strong rules for discarding predictors in lasso-type problems," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 245–266, 2012.

[131] A. Tipping and A. Faul, "Analysis of sparse Bayesian learning," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2002, pp. 383–390.

[132] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[133] M. Tipping, A. Faul *et al.*, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1, no. 3, 2003.

[134] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[135] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Processing*, vol. 86, pp. 589–602, 2006.

[136] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing*, vol. 86, pp. 572–588, 2006.

[137] E. Van Den Berg and M. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[138] N. Vaswani, "LS-CS-residual (LS-CS): compressive sensing on least squares residual," *Signal Processing, IEEE Transactions on*, vol. 58, no. 8, pp. 4108–4120, 2010.

[139] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *Signal Processing, IEEE Transactions on*, vol. 58, no. 9, pp. 4595–4607, 2010.

[140] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. of the15th IEEE International Conference on Image Processing (ICIP 2008)*, San Diego, USA, 2008, pp. 893–896.

[141] J. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *arXiv:1207.3107*, 2012.

[142] K. Walhovd, A. Fjell *et al.*, "Multi-modal imaging predicts memory performance in normal aging and cognitive decline," *Neurobiol Aging*, vol. 31, no. 7, pp. 1107–1121, 2010.

[143] J. Wan, Z. Zhang, J. Yan, T. Li, B. Rao, S. Fang, S. Kim, S. Risacher, A. Saykin, and L. Shen, "Sparse Bayesian multi-task learning for predicting cognitive outcomes from neuroimaging measures in Alzheimer's disease," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 940–947.

[144] H. Wang *et al.*, "A new sparse multi-task regression and feature selection method to identify brain imaging predictors for memory performance," *ICCV 2011*, pp. 557–562.

[145] Y. Wang and S. Makeig, "Predicting intended movement direction using eeg from human posterior parietal cortex," *Lecture Notes in Computer Science*, vol. 5638, pp. 437–446, 2009.

[146] Z. Wang and A. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[147] M. W. Weiner, P. S. Aisen *et al.*, "The alzheimer's disease neuroimaging initiative: progress report and future plans," *Alzheimers Dement*, vol. 6, no. 3, pp. 202–11 e7, 2010.

[148] D. Wipf, S. Nagarajan *et al.*, "A unified bayesian framework for MEG/EEG source imaging," *Neuroimage*, vol. 44, no. 3, pp. 947–966, 2009.

[149] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 317–329, 2010.

[150] D. P. Wipf, "Sparse estimation with structured dictionaries," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 2016–2024.

[151] D. Wipf, "Bayesian methods for finding sparse representations," *Ph.D. Thesis, University of California, San Diego*, 2006.

[152] D. Wipf, J. Owen, H. Attias, K. Sekihara, and S. Nagarajan, "Robust Bayesian estimation of the location, orientation, and time course of multiple correlated neural sources using meg," *NeuroImage*, vol. 49, no. 1, pp. 641–655, 2010.

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

[154] ——, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3704–3716, 2007.

[155] D. Wipf, B. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6236–6255, 2011.

[156] J. Wright, A. Y. Yang, A. Ganesh, and et al, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[157] Y. Yang *et al.*, "Tag localization with spatial correlations and joint group sparsity," in *CVPR 2011*, pp. 881–888.

[158] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse bayesian inference," *to appear in IEEE Transactions on Signal Processing*, vol. PP, no. 99, p. 1, 2012.

[159] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Processing*, vol. 92, no. 1, pp. 259–269, 2012.

[160] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, pp. 49–67, 2006.

[161] D. Zachariah, S. Chatterjee, and M. Jansson, "Dynamic iterative pursuit," *Signal Processing, IEEE Transactions on*, vol. 60, no. 9, pp. 4967–4972, 2012.

[162] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Bayesian pursuit algorithm for sparse representation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009, pp. 1549–1552.

[163] R. Zdunek and A. Cichocki, "Improved M-FOCUSS algorithm with overlapping blocks for locally smooth sparse signals," *IEEE Trans. on Signal Processing*, vol. 56, no. 10, pp. 4752–4761, 2008.

[164] D. Zhang and D. Shen, "Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease," *Neuroimage*, vol. 59, no. 2, 2012.

[165] Z. Zhang and B. Rao, "Recovery of block sparse signals using the framework of block sparse bayesian learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 3345–3348.

[166] Z.-L. Zhang and Z. Yi, "Robust extraction of specific signals with temporal structure," *Neurocomputing*, vol. 69, no. 7-9, pp. 888–893, 2006.

[167] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of non-invasive fetal ECG via block sparse Bayesian learning," *IEEE Trans. on Biomedical Engineering, accepted*.

[168] ——, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Trans. on Biomedical Engineering, accepted*.

[169] ——, "Spatiotemporal sparse Bayesian learning with applications to compressed sensing of multichannel ECG for wireless telemonitoring," *submitted to IEEE Trans. on Biomedical Engineering*, 2012.

[170] Z. Zhang and B. D. Rao, "Exploiting correlation in sparse signal recovery problems: Multiple measurement vectors, block sparsity, and time-varying sparsity," in *ICML 2011 Workshop on Structured Sparsity: Learning and Inference*.

[171] ——, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. on Signal Processing, accepted*.

[172] ——, "Sparse signal recovery in the presence of correlated multiple measurement vectors," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 3986–3989.

[173] ——, "Iterative reweighted algorithms for sparse signal recovery with temporally correlated source vectors," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 3932–3935.

[174] ——, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.

[175] Z. Zhang, J. Wan, B. D. Rao, S. Fang, A. Saykin, and L. Shen, "Correlation- and nonlinearity-aware sparse Bayesian learning with applications to the prediction of cognitive scores from neuroimaging measures," in *submitted to 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[176] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 814–822.

[177] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *arXiv preprint arXiv:1205.4080*, 2012.

[178] ——, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Trans. on Signal Processing*, 2012.

[179] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Proc. of the 44th Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 808–812.

[180] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.