Multi-task Learning of Gaussian Graphical Models for Brain Connectivity by Subject Variability

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multi-task learning of Gaussian graphical models has been applied to jointly learn the brain functional connectivity for several subjects [11] [20]. However, existing multi-task learning algorithms treat all subjects equivalently. Often in BCI data collected from different subjects may vary a lot due to many factors (e.g. structured noise). Treating subjects equivalently sometimes would lower the performance. In this paper we study the problem of multi-task learning for Gaussian graphical models while considering the variability across subjects. We formulate the problem in hierarchical Bayesian model and provide an alternating technique to the optimization problem. The data from simulation and the real world validate the performance of our method.

1 Introduction

The brain functional connectivity graphs are of high neuroscientific interest since their structures reflect fundamental nervous system assembly principles. Gaussian graphical models have been used to identify the functional connectivity which helps the effective diagnosis of Alzheimer's disease [12]. Multi-task learning is able to more robustly estimate the brain connectivity by considering the shared structure information between tasks. In our semantic, each single task is estimating the brain connectivity using a Gaussian graphical model for a subject. Multi-task learning of Gaussian graphical models based on $\ell_{1,\infty}$[11] and $\ell_{2,1}$ [20] regularizers learns consistent structures across subjects. However, current multi-task learning algorithms treat subjects equivalently without considering the subject variability. Here we assume the subject variability comes in two ways: 1. the small difference in brain function connectivity among subjects under some specific task even though they share major connectivity patterns; 2. overlapped structured noise, which can result from a lot of factors, like artifacts, the location of electrodes, mental and emotional states and so on. Our model in the following incorporates the first case. However, here we put more focus on the second case, which is not shared by other multi-task learning algorithms. In many cases, ignoring subject variabilities in the second case could possibly result in much worsened results. This can be easily understood by simply considering the extreme case in which one of the datasets is an outlier. We simply denote the subject variability in the second case as the data quality variability. In this paper, we learn the brain functional connectivity using Gaussian graphical models by considering the variability of data quality among subjects. Specifically, in our hierarchical Bayesian model the data quality of subjects are automatically inferred and thus subjects are treated differently. For example, noisier subjects would contribute less to the share structure information.

A Gaussian graphical model is a probabilistic graph which assumes all the variables are continuous and following multivariate normal distribution. Let $X = (X^{(1)}, X^{(2)}, ..., X^{(p)})$ be p dimensional random variables which have a multivariate Gaussian distribution with mean μ and covariance Σ. Given N training samples $X_1, X_2, ..., X_N$, we want to estimate the concentration matrix Ω (which is the inverse of the covariance matrix Σ). If the $ijth$ component of Ω is zero, then $X^{(i)}$ and $X^{(j)}$ are conditionally independent given all other variables.

Structure learning for a graphical model is to find the topology of the graph which can accurately describe the given samples while maintains low complexity in order to have better generalization. For Gaussian graphical models, structure learning is to learn the parameters and identify zeros in

the concentration matrix (also known as covariance selection). In the corresponding graph, the edge between the variables $X^{(i)}$ and $X^{(j)}$ is missing if and only if Ω_{ij} equals to zero. To encourage the sparseness of the structures, several methods have been proposed, among which the most popular one is maximizing the log-likelihood while enforcing a ℓ_1 penalty on Ω [5, 22, 8, 14].

Multi-task learning investigates the challenge of sharing information among related tasks. To discover shared characters of related tasks, the multi-task model often assumes shared regularization [7, 11] or a shared prior [21, 2]. Following the second philosophy, in this paper we consider hierarchial Bayesian modeling for the multi-task learning of Gaussian graphical models. By assuming that individual model parameters are drawn from a common hyperprior, the hierarchial Bayesian model provides a natural and effective way to regularize individual models. But different from the traditional formulation, to consider the data quality variability between tasks, we do not assume the parameters of each task are drawn from the exactly same hyperprior. We need a factor in the hyperprior to account for the variability of data quality among tasks. To serve this purpose, we consider the inverse Wishart distribution as the prior of the covariance of each task. The inverse scale matrix of the inverse Wishart distribution captures the extracted information from different tasks while the degree of freedom for each task controls the extent that the data structure of individual task varies from the inverse scale matrix. Even though the inverse-Wishart distribution has been widely used as the conjugate prior for covariance matrix, as far as we know there is no work in multi-task learning which simultaneously estimates the inverse scale matrix as the share information and the degree of freedom for each task. We derive an iterating algorithm to estimate the parameters. The optimization used in the iterations can be solved with off-the-shelf package.

The remainder of this paper is organized as follows. In section 2, we give a brief introduction to the learning of Gaussian graphical models. In section 3, we formulate the multi-task learning of Gaussian graphical models in the hierarchial Bayesian framework and develop the iterating learning algorithm. Section 4 contains the experimental results which validate our model with both simulation data and real world data. And we conclude our work in section 5.

1.1 Related Work

Recently multi-task learning has been an active research field in machine learning and has been applied to a diverse of problems. Lawrence & Platt [15] and Yu et al. [21] present different multi-task approaches for Gaussian Processes. The former learns the parameters of a shared covariance while the latter assumes the model parameters share the same prior. Similarly, Alamgir et al. [2] learn linear classifiers for BCI by assuming the parameters of different tasks are drawn from the one multivariate normal distribution. Obozinski & Taskar [18] assume that variables are sparsely related to prediction and the sparsity to be shared among tasks. Argyriou et al. [3] assume the sharing subspace in the predictor space. Zhang & Yeung [23] derive a convex formulation for multi-task learning by taking advantage of a matrix-variate normal distribution as the prior for model parameters.

The work of Honorio & Samaras [11] presents a multi-task learning method specifically for the Gaussian graphical models through the $l_{1,\infty}$ norm regularization. Similar approach based on $\ell_{2,1}$ regularizer was proposed by Varoquaux et al. [20]. Both approaches emphasize on enforcing the same sparsity among concentration matrices. However, these methods treat subjects equivalently and are suboptimal when the data quality is uneven among subjects.

The inverse-Wishart distribution is commonly used as the conjugate prior of the population covariance in the multivariate normal distribution in Bayesian analysis. In the research of Gaussian graphical models, it has already been used as the prior for Bayesian covariance selection. The commonly used prior for covariance selection is $\Sigma \sim IW(\delta, \tau I)$ which was called "conventinal proper prior" by Berger & Pericchi [6]. As the conjugate prior of covariance, it was also widely used to compute marginal likelihoods [10, 13, 4]. The inverse-Wishart distribution was also used in the multi-task learning of Gaussian process [21]. But in most of application listed here, the inverse-Wishart distribution was only used as the noninformative prior. Neither the scale matrix was learned to store the shared information nor degree of freedom was used to control task variability.

2 Learning Gaussian Graphical Models

Given random samples $X_1, X_2, ..., X_n$, the likelihood of concentration matrix Ω is

$$\log |\Omega| - tr(\Omega S) \tag{1}$$

where $|\cdot|$ denotes the determinate of a matrix; tr denotes the trace of a matrix and S is the sample covariance matrix:

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})'. \tag{2}$$

However, simply maximizing the likelihood does not automatically lead to sparse Ω since the elements in Ω are typically nonzero. To encourage the sparseness of Ω, the ℓ_1 norm regularization was introduced. Then the object function to be maximized becomes

$$\log|\Omega| - tr(\Omega S) - \rho \parallel \Omega \parallel_1 . \tag{3}$$

Several approaches have been proposed to solve this problem: a method by Yuan & Lin [22] which takes advantage of the maxdet algorithm [19]; the solution [5] formulated as box-constrained quadratic programming solved with interior point procedure and the graphical lasso by Friedman et al. [8] which solves the convex dual by lasso estimates; Instead of directly solving eq. (3), Meinshausen et al. [17] approximates the conditional dependencies by performing lasso regression for each variable. Recently Krishnamurthy & d'Aspremont [14] solve problem (3) by formulating a pathwise algorithm, which is then solved by iterations consisting of solving structured linear system and improving precision by a coordinate descent algorithm.

3 Learning from Multiple Tasks

In the previous section, we have introduced the learning algorithms that learn a single Gaussian graphical model given data D. In contrast, here we estimate the structures of m related Gaussian graphical models simultaneously based on training data $D_{1:m}$ while considering the shared structure information among them. Following the hierarchial Bayesian modeling, we assume that the covariance matrix $\Sigma_{1:m}$ are sampled from $P_{G,\nu_{1:m}}$ where G represents the shared structure information and $\nu_{1:m}$ controls the variability of tasks. For the learning, we not only need to learn covariance matrices $\Sigma_{1:m}$ (or $\Omega_{1:m}$) but also the common prior $P_{G,\nu_{1:m}}$. In the following subsections, we first discuss the Inverse-Wishart Distribution; We then present the probabilistic framework for multi-task learning and the iteration learning algorithm.

3.1 Inverse-Wishart Distribution

In this section we discuss the inverse-Wishart distribution which will serve as the prior for the covariance matrix Σ and some simple properties for the convenience of the following discussion. Consider the covariance matrix Σ with size $p \times p$. Σ is said to follow the inverse Wishart distribution $IW_p(G, \nu)$ with scale matrix G and degree of freedom ν ($\nu > p-1$) if [9]:

$$p(\Sigma|G,\nu) = \frac{|G|^{\nu/2}|\Sigma|^{-(\nu+p+1)/2}\exp(-tr(G\Sigma^{-1})/2)}{2^{\nu p/2}\Gamma_p(\nu/2)}. \tag{4}$$

The mean of Σ is given by Mardia et al. [16]:

$$E(\Sigma) = \frac{G}{\nu - p - 1}, \tag{5}$$

and the variance of each element of Σ is given by:

$$var(\Sigma_{ij}) = \frac{(\nu-p+1)G_{ij}^2 + (\nu-p-1)G_{ii}G_{jj}}{(\nu-p)(\nu-p-1)^2(\nu-p-3)}. \tag{6}$$

To make the mean of Σ unaffected by ν, we reparametrize the inverse Wishart distribution as:

$$\Sigma|G,\nu \sim IW_p((\nu-p-1)G,\nu). \tag{7}$$

Then the mean and variance of Σ become:

$$E(\Sigma) = G, \tag{8}$$

$$var(\Sigma_{ij}) = \frac{(\nu-p+1)G_{ij}^2 + (\nu-p-1)G_{ii}G_{jj}}{(\nu-p)(\nu-p-3)}. \tag{9}$$

From (9) we observe that the degree of freedom ν controls how much Σ varies from the prior structure G.

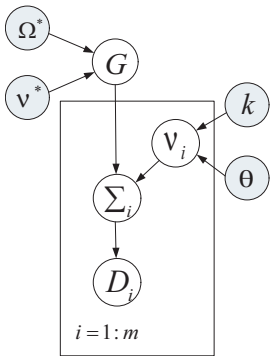

Figure 1: Illustration of Hierarchical Bayesian Modeling for Multi-task Gaussian Graphical Modeling. The shaded nodes represent hyper parameters that are fixed

3.2 Probabilistic Framework

The hierarchial Bayesian modeling of multi-task learning of Gaussian graphical models is illustrated in figure 1. Parameter G defines the shared prior structure by $\Sigma_{1:m}$. And the learned G captures the extracted information from $\Sigma_{1:m}$. The degree of freedom ν_i controls the extent that each Σ_i deviates from G. In the following analysis, we will see that from the perspective of learning G, ν_i forms the weight with which G selectively learns information from Σ_i. From another point of view, each task does not contribute equally to the shared structure information, which is dramatically different from traditional multi-task learning approaches. The parameter G is given the inverse Wishart prior $IW_p(\Omega^*, \nu^*)$ with scale matrix Ω^* and the degree of freedom ν^*. The prior for ν_i is the Gamma distribution $Gamma(k, \theta)$ with shape parameter k and scale parameter θ.

In the model shown in figure 1 we need to estimate parameters G, $v_{1:m}$ and $\Omega_{1:m}$ ($\Omega_k = \Sigma_k^{-1}$) given data from multi-tasks $D_{1:m}$. Instead of estimating the concentration matrix by maximum likelihood as in the single task learning, the estimation is conducted by maximizing the posterior distribution of G, $v_{1:m}$ and $\Omega_{1:m}$ given data $D_{1:m}$. Our object function is formulated as:

$$\begin{aligned} \max_{G,\nu_{1:m},\Omega_{1:m}} \quad & \log P(G, \nu_{1:m}, \Omega_{1:m}|D_{1:m}) - \rho \sum_{k=1}^{m} \|\Omega_k\|_1. \\ s.t. \quad & \Omega_{1:m} \succ 0, \quad G \succ 0, \quad \nu_{1:m} > p - 1 \end{aligned}$$

where $\succ$ means positive definite and the ℓ_1 term enforces the sparseness of $\Omega_{1:m}$. Following the graphical structure in Figure 1, the first term of the object function in equ.(10) can be factorized as:

$$\begin{aligned} & \log P(G, \nu_{1:m}, \Omega_{1:m}|D_{1:m}) \\ \propto \quad & \log P(G, \nu_{1:m}) P(\Omega_{1:m}|G, \nu_{1:m}) P(D_{1:m}|\Omega_{1:m}) \\ \propto \quad & \log P(G) + \sum_{i=1}^{m} [\log P(\nu_i) + \log P(\Omega_i|G, \nu_i) + \log P(D_i|\Omega_i)], \end{aligned} \tag{10}$$

where $P(G) = P(G^{-1}) \sim IW_p(\Omega^*, \nu^*)$, $P(\nu_i) \sim Gamma(k, \theta)$ and

$$\log P(D_k|\Sigma_k) = -\frac{N_k}{2} \left(p \log(2\pi) - \log|\Omega_k| + tr(\Omega_k S_k)\right) \tag{11}$$

where N_k is the number of samples in the task k and S_k is defined as in equ.(2) for the task k. Note that the term N_k is usually dropped in equ. (1) which was used by single task learning. But for multi-task learning, we need to combine the likelihoods of different tasks into one object function. Thus it is important to keep this term.

Combining with equ. (4) (10) and (11), the object function in equ. (10) can be further detailed as:

$$
\begin{aligned}
& \max_{\Omega_{1:m},G,\nu_{1:m}} \sum_{k=1}^{m}\{-\frac{\nu_k p}{2}\log 2 - \log\Gamma_p(\nu_k/2) + (\frac{N_k}{2}+\frac{\nu_k+p+1}{2})\log|\Omega_k| - \frac{N_k}{2}tr(\Omega_k S_k) \\
- \quad & \frac{1}{2}tr((\nu_k-p-1)G\Omega_k) + \frac{\nu_k}{2}\log|(\nu_k-p-1)G| - \rho\|\Omega_k\|_1 + \log(k-1)\nu_k - \frac{\nu_k}{\theta}\} \\
+ \quad & \frac{\nu^*}{2}\log|(\nu^*-p-1)\Omega^*| + \frac{\nu^*+p+1}{2}\log|G| - \frac{1}{2}tr((\nu^*-p-1)\Omega^* G) \\
s.t. \quad & \Omega_{1:m} \succ 0, \quad G \succ 0, \quad \nu > p-1.
\end{aligned}
$$

3.3 Iterating Algorithm

Theorem 1 For the object function in equ.(12), we have
(i) Given G and ν, (12) is concave with respect to $\Omega_{1:m}$.
(ii) Given $\Omega_{1:m}$ and ν, (12) is concave with respect to G.
Proof The constraints in equ.(12) are convex with respect to all variables and let us see the object function for each case. (i): Except for the term $-\frac{1}{2}tr((\nu_k-p-1)G\Omega_k)$, the convexity of terms containing Ω_k in (12) can be easily explained by using similar argument that proves the convexity of the likelihood of Ω in equ.(1). The term $-\frac{1}{2}tr((\nu_k-p-1)G\Omega_k)$ is a linear function of Ω_k. Thus it is both convex and concave with respect to Ω_k. Because the summation operation preserves convexity, (i) is proved. (ii) can be proved by noticing that $\log|G|$ is concave with respect to G.

Based on the observation of the convexity properties in theorem 1, we develop an alternating method to solve the problem and discuss some properties. Specifically, the alternating algorithm consists of three steps and in each step we solve one parameter by holding the other two parameters. This procedure is repeated until converge. In the following, we will present the three iterating steps in detail.

3.3.1 Solve $\Omega_{1:m}$ by holding G and ν

If G and ν are fixed, $\Omega_{1:m}$ are independent of each other in equ.(12). Thus the optimization of $\Omega_{1:m}$ can be done separately for each Ω_k. Specifically, for $k = 1 : m$ we solve Ω_k by the following optimization problem:

$$
\begin{aligned}
\max_{\Omega_k} \quad & (\frac{N_k}{2}+\frac{\nu_k+p+1}{2})\log|\Omega_k| - \frac{1}{2}tr(\Omega_k(N_k S_k + (\nu_k-p-1)G)) - \rho\|\Omega_k\|_1 \\
s.t. \quad & \Omega_k \succ 0.
\end{aligned} \tag{12}
$$

In the object function of equ.(12) we drop some constants that are not related to Ω_k. The formulation of (12) is much like that of the corresponding single task learning in equ.(1) except for the additional G in the second term. It needs to be emphasized that G plays an important role in (12) because with the term G, the learned Ω_k incorporates not only the information from the task k but also the shared information from other tasks. Here we solve it using the cvx [1] package.

3.3.2 Solve G by holding $\Omega_{1:m}$ and ν

In this case, the optimization problem in equ.(12) can be simplified as:

$$
\begin{aligned}
\min_{G} \quad & \sum_{k=1}^{m}[tr((\nu_k-p-1)G\Omega_k) - \nu_k\log|(\nu_k-p-1)G|] \\
& -(\nu^*+p+1)\log|G| + tr((\nu_k-p-1)\Omega^* G) \\
s.t. \quad & G \succ 0.
\end{aligned} \tag{13}
$$

To simplify the problem, we first relax the positive definite constraint in equ.(13) as semi-positive definite: $G \succeq 0$. And then we change the constraint $G \succeq 0$ into its equivalent form $y^T G y \geq 0$. Because both the object function and the feasible region are convex with respect to variable G, by KKT condition we have

$$
\begin{cases}
\sum_{k=1}^{m}\left[(\nu_k-p-1)\Omega_k - \nu_k G^{-1} - \lambda y y^T\right] \\
\qquad -(\nu^*+p+1)G^{-1} + (\nu_k-p-1)\Omega^* = 0 \\
y^T G y \geq 0 \\
\lambda y^T G y = 0 \\
\lambda \geq 0.
\end{cases} \tag{14}
$$

[1] http://cvxr.com/cvx/

It is not difficult to see that $\lambda = 0$ and

$$G = \left[\frac{1}{\nu^* + p + 1 \sum_{k=1}^{m} \nu_k} \left[(\nu^* - p - 1)\Omega^* + \sum_{k=1}^{m} (\nu_k - p - 1)\Omega_k \right] \right]^{-1} \tag{15}$$

are the solution of equ.(14). Since it is a convex programming problem, G in (15) is also the solution of (13) with relaxed constraints. Further since $\Omega_{1:m}$ are positive definite, G in (15) is positive definite. Thus (15) is also the solution of (13).

Equ. (15) holds the central ideal that the shared structure G is not evenly extracted from tasks but weighted by the factor $\nu_k - p - 1$. And it is expected that after learning tasks with good data quality would result in greater ν_k, thus contributing more information to G. Conversely, tasks with low data quality would have smaller ν_k, weighted less in the shared information.

3.3.3 Solve $\nu_{1:m}$ by holding G and $\Omega_{1:m}$

With G and $\Omega_{1:m}$ fixed, equ.(12) is neither convex nor concave with respect to $\nu_{1:m}$. But we also notice that by fixing G and $\Omega_{1:m}$, $\nu_1, ..., \nu_m$ are independent. So we can solve each of them separately and this can be done easily with conjugate gradient by noticing that $\Gamma_p(u) = \pi^{\frac{p(p-1)}{4}} \prod_{j=1}^{p} \Gamma(u + (1-j)/2)$ and $\nabla_u \log \Gamma(u) = -\gamma - \frac{1}{u} + \sum_{n=1}^{\infty} \left(\frac{1}{n} - \frac{1}{u+n} \right)$ where the constant $\gamma \approx 0.577215$ [1].

We solve $\Omega_{1:m}$, G and ν by iterating through sections (3.3.1), (3.3.2) and (3.3.3). The matrix G is initialized using equ.(8) where $E(\Sigma)$ is replaced with the mean of sample covariances.

3.4 Selection of the tuning parameter ρ

Up to now we have been focusing on solving the problem in equ.(12) by assuming that the tuning parameter ρ is fixed. The selection of ρ is an important problem but it is not our focus here. Considering the computational efficiency, in the experimental comparison in the next section, we just apply BIC (Bayesian information criterion) score for the selection of ρ which was also used by Yuan & Lin [22]. But for multi-task learning we need to estimate the goodness of ρ for all the tasks and we slightly adjust the standard BIC sore into the semantic of multi-task learning which is simply the average of the scores used in the single task learning:

$$BIC_{MTL}(\rho) = \frac{1}{m} \sum_{k}^{m} - \log |\Omega_k(\rho)| + tr(\Omega_k(\rho) S_k) + \frac{\log(N_k)}{N_k} \sum_{i \leq j} e_{ij}^k(\rho), \tag{16}$$

where $e_{ij}^k(\rho) = 0$ if $\Omega_{k(ij)} = 0$ and otherwise $e_{ij}^k(\rho) = 1$

4 Experiments

In this section we validate our model on both synthetic data and real data. The method proposed is compared with three other models: 1)STL: the single task learning which learns a Gaussian graphical model for each task by using graphical lasso. 2)MT: the exactly same model as we proposed but we fix the same ν for all tasks. This comparison can demonstrate the benefits of considering the data quality variability between tasks over treating tasks equivalently. 3)SM: a Gaussian graphical model which was learned with the graphical lasso using the stacked data from all tasks. In this case it is assumed that all tasks share a single model. Our current multi-task learning with separate ν for each task as MTSV. Different models are compared by assessing quantitatively the goodness of fit of the inferred connectivity model to new data. For this purpose, all the experiments in the following are based on five fold cross validation which tests the generalization power of the learned model.

For implementation, the hyperprior parameter Ω^* is directly calculated from training data: $\Omega^* = \left[\frac{1}{m} \sum_{k=1}^{m} S_k \right]^{-1}$ and we set $\nu^* = p + 2$.

4.1 On the Simulation Data

To generate data for different tasks among which some data structures are shared, we first generate a parent Gaussian graphical model with the concentration matrix Ω. Then the concentration matrix of each task is sampled from the parent model by randomly creating some different links. In this

experiment the randomly generated matrix Ω is

$$\Omega = \begin{bmatrix} 7 & 0 & 1 & 0 & 1 & 2 \\ 0 & 9 & 0 & 2 & 0 & 0 \\ 1 & 0 & 11 & 3 & 0 & 3 \\ 0 & 2 & 3 & 8 & 3 & 3 \\ 1 & 0 & 0 & 3 & 8 & 0 \\ 2 & 0 & 3 & 3 & 0 & 10 \end{bmatrix}. \tag{17}$$

For each task k, we sample its concentration matrix Ω_k from Ω in this way: for each element in the upper triangular part (excluding the diagonal elements) of Ω, we have pt percentage of chance to change the element and (100-pt) percentage to remain the element same in Ω_k. When the element is chosen to change, if the element in Ω is not zero, we change it to zero in Ω_k. Otherwise, we change the zero element to a value uniformly generated between 1 and 3 (to make sure that Ω_k is still positive definite). The diagonal elements are shared by Ω and Ω_k. Then we fill the lower triangular of Ω_k by making it symmetric. The data D_k is then sampled from the Gaussian graphical model with concentration matrix Ω_k.

We also want to simulate the tasks with low data quality. Those tasks with low data quality are assumed to be corrupted by structured Gaussian noise. The structured Gaussian noise R is sampled from $N(0, aa^T)$ where $a \in R^{p \times p}$ and each element of a is sampled from uniform distribution $U[0, 1]$.

In this experiment, pt was set as 20 and data for eight tasks $D_{1:8}$ are generated using the procedure as stated above. And the size of each data set is 20. Then D_6, D_7, D_8 are corrupted by Gaussian noise in the way:

$$\begin{cases} D_6 \leftarrow 0.8D_6 + 0.2R \\ D_7 \leftarrow 0.5D_7 + 0.5R \\ D_8 \leftarrow R. \end{cases} \tag{18}$$

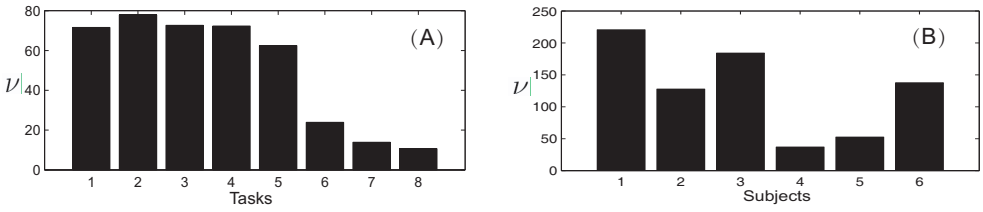

Figure 2: (A): Learned ν for each simulated task; (B)Learned ν for EEG data

Figure 2 (A) shows the learned parameter ν for each task. It is obvious that the model can identify the data quality of different tasks because the first five noise free tasks get great value of ν while the ν for the last three tasks are closely related to the amount of noise added as in (18).

As shown in Table 1 (A), overall the multi-task learning (MTSV) with separate ν for each task achieves the best results over these four methods. Disturbed by noisy tasks $T_{6:8}$, multi-task learning (MT) does not seem obviously better than single task learning (STL). By comparison with MT, the performance of MTSV on $T_{1:5}$ are obviously less affected by the last three noisy data sets. There is no need to compare results on T_8 which is completely noise.

Table 1: Log-likelihood of structures learned by four approaches (A) Simulated data; (B) EEG data

(A)	T1	T2	T3	T4	T5	T6	T7	T8
MTSV	**5.9**	**6.1**	**6.0**	**6.0**	**6.1**	**2.6**	**-6.9**	-21.2
MT	5.2	5.4	5.4	5.3	5.5	1.7	-8.3	-22.7
STL	5.1	5.3	5.4	4.9	3.5	1.0	-8.1	-14.3
SM	1.9	2.0	2.0	2.0	2.1	0.9	-7.5	-21.1

(B)	S1	S2	S3	S4	S5	S6
MTSV	**21.1**	**23.3**	**21.9**	**5.1**	**4.9**	**19.1**
MT	19.9	21.2	20.6	1.8	0.3	17.7
STL	19.5	20.4	19.4	4.7	4.3	16.9
SM	16.4	18.1	17.4	-1.6	-4.2	14.9

4.2 On the EEG Data

We used the EEG dataset that captures the brain activity of a subject who was operating a uninhabited air vehicle(UAV) simulator. The UAV was equipped with bombs and flew through a preplanned pathway. In order to destroy the targets which were embedded in background noise, the operator had to designate the target with a mouse, select and release weapons.

EEG data were collected from the 19 electrode sites with the 10-20 system. The EEG signals recorded from the electrodes placed on the mastoids were used as the ground and reference for signals from other EEG electrodes. The data were sampled at 256Hz. We first re-reference the raw EEG signals by common average reference (CAR) filter which subtract $\frac{1}{H}\sum_{q=1} Hs_q$ from each channel, where H is the total number of channels and s_q is the collected signal at the qth channel and at the particular time.

We collect EEG data from six subjects and each subject with 200 samples. And we try to learn a Gaussian graphical model for each subject, which can be further used for exploiting the function relationship between different parts of the brain or possibly estimating the workload of subjects. Considering the number of available data and the quality of EEG signals, we select 14 channels out of 19 for the experiment.

Figure 2(B) is the learned degree of freedom for each subject from which we can easily identify that the data quality of subject 4 and 5 are not as good as others. Table 1 (B) lists the log likelihood of all methods for each subject. Overall, MTSV achieves the best results. And both MTSV and MT get better results than STL.

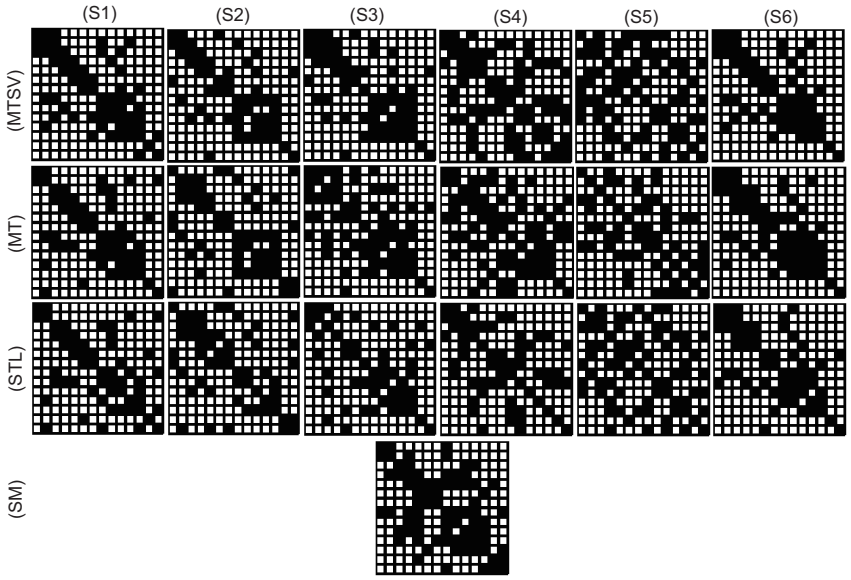

Figure 3: Learned connection patterns for six subjects using four methods in which the black dot means its corresponding row and column variables are connected

Figure 3 shows the learned connection structures for the six subjects with four methods. Obviously the data sets of S4 and S5 seem to be much noisier than those of S1, S2, S3 an S6. The structures of S1, S2, S3 and S6 learned by MTSV are more consistent than those by MT and STL. The 4 structures (S1, S2, S3 and S6) learned by MT seem to be affected by the noise from S4 and S5, which is most obvious for S3. The structures learned by STL vary too much across subjects. And the structure learned by SM does not seem to fit any subject well.

5 Discussion

In this paper, we presented a multi-task learning of Gaussian graphical model for the brain functional connectivity with special consideration to data quality variability between subjects. An iteration algorithm was developed to learn the model where each iteration is an optimization problem which can solved by using regular optimization packages. The experiments validate the effectiveness of our model.

The proposed method is possible to scale to large problems. The major computation required of the algorithm is in the first iteration (equ.12). However, we can observe that the difference between equ.12 and the standard formulation (equ.3), which has been solved efficiently by graph lasso, is only the in the values of coefficients. So it should not be difficult to adjust graph lasso method to solve equ.12 efficiently for large problems.

There are several ways to extend current work. In this paper we assume same regularization parameter ρ for all the tasks. But it is not necessary for all tasks sharing the same sparsity. Thus how to adjust ρ for each task needs to be further investigated. Even though the iteration algorithm developed converge in a small number of steps, the theoretical analysis of convergence needs to be performed.

References

[1] Abramowitz, M. and Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972.

[2] Alamgir, Morteza, Grosse-Wentrup, Moritz, and Altun, Yasemin. Multitask learning for brain-computer interface. In *AISTATS*, 2010.

[3] Argyriou, Andreas, Evgeniou, Theodoros, Pontil, Massimiliano, Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex multi-task feature learning. In *Machine Learning*. press, 2007.

[4] Atay-Kayis, Aliye and Massam, Helandegrave;ne. A monte carlo method for computing the marginal likelihood in nondecomposable gaussian graphical models. *Biometrika*, 92(2):317–335, June 2005.

[5] Banerjee, Onureena, Ghaoui, Laurent El, and Natsoulis, Georges. Convex optimization techniques for fitting sparse gaussian graphical models. In *ICML*, pp. 89–96. Press, 2006.

[6] Berger, J.O. and Pericchi, L. Objective bayesian methods for model selection: introduction and comparison. *Institute of Mathematical Statistics Lecture Notes- Monograph Series*, 38: 135–207, 2001.

[7] Evgeniou, Theodoros, Micchelli, Charles A., and Pontil, Massimiliano. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[8] Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008.

[9] Gelman, Andrew, Carlin, John B., Stern, Hal S., and Rubin, Donald B. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, July 2003. ISBN 158488388X.

[10] Giudici, Paolo and Green, Peter J. Decomposable graphical gaussian model determination, 1999.

[11] Honorio, Jean and Samaras, Dimitris. Multi-task learning of gaussian graphical models, 2010.

[12] Huang, Shuai, Li, Jing, Sun, Liang, Ye, Jieping, Fleisher, Adam, Wu, Teresa, Chen, Kewei, and Reiman, Eric. Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 2010.

[13] Jones, Beatrix, Carvalho, Carlos, Dobra, Adrian, Hans, Chris, Carter, Chris, and West, Mike. Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science*, 20:388–400, 2004.

[14] Krishnamurthy, Vijay and d'Aspremont, Alexandre. A pathwise algorithm for covariance selection. 2009.

[15] Lawrence, Neil D. and Platt, John C. Learning to learn with the informative vector machine. In *ICML*. Morgan Kaufmann, 2004.

[16] Mardia, Kanti V., Kent, J. T., and Bibby., J. M. *Multivariate Analysis*. Academic Press, 1979. ISBN 0-12-471250-9.

[17] Meinshausen, Nicolai, B??hlmann, Peter, and Z??rich, Eth. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[18] Obozinski, Guillaume and Taskar, Ben. Multi-task feature selection. Technical report, Workshop in ICML, 2006.

[19] Vandenberghe, Lieven, Boyd, Stephen, and po Wu, Shao. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19: 499–533, 1998.

[20] Varoquaux, Gaël, Gramfort, Alexandre, Poline, Jean Baptiste, and Thirion, Bertrand. Brain covariance selection: better individual functional connectivity models using population prior. In Zemel, Richard and Shawe-Taylor, John (eds.), *NIPS*, Vancouver, Canada, December 2010. John Lafferty.

[21] Yu, Kai, Tresp, Volker, and Schwaighofer, Anton. Learning gaussian processes from multiple tasks. In *ICML*, pp. 1012–1019, 2005.

[22] Yuan, Ming and Lin, Yi. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, March 2007.

[23] Zhang, Yu and Yeung, Dit-Yan. A convex formulation for learning task relationships in multi-task learning. *UAI*, July 2010.