Encoding Low-Rank and Sparse Structures Simultaneously in Multi-task Learning

Shike Mei
Tsinghua University
meiskoier@gmail.com

Bin Cao
Microsoft Research Asia
bincao@microsoft.com

Jiantao Sun
Microsoft Research
jtsun@microsoft.com

Abstract

Multi-task learning (MTL) aims to improve the performance of each task by borrowing the knowledge learned from other related tasks. Identifying the underlying structures among tasks is crucial for MTL to understand the relationship among tasks. In this paper, we propose a novel multi-task learning model to simultaneously consider low-rank structure and sparse structure. Combining these two types of structures could not only improve the learner's performance, but also make the interpretation of learned structures easier. However, the standard subgradient optimization method for solving this problem could only achieve a rate of convergence $O(1/\sqrt{k})$. We propose a novel optimization method combining the Moreau approximation and an accelerated proximal method to achieve a rate of convergence $O(1/k)$. We conduct experiments on synthetic data and several real-world data sets and the results show the gains of our model in comparison with state-of-the-art baselines.

1 Introduction

Multi-task learning (MTL) aims to improve the generalization performance of each learning task by exploiting the intrinsic structures shared among a group of tasks. The shared structures, which encode the common knowledge of related tasks, are crucial for the success of multi-task learning. MTL is especially desirable when many related tasks are required for learning but the training data for each of them is limited. With the help of the learnt structures of multiple tasks, the scarcity of training data can be alleviated. In previous studies, different structures were assumed to exist among tasks and corresponding algorithms are proposed to learn such structures. For example, [10] assumes that all tasks are related to each other, [25, 24, 1] assume a shared low-dimension feature space among tasks, [26] uses a covariance matrix to model the relationship among tasks. [11] assumes a cluster structure shared among tasks.

Two types of structures have been extensively studied in MTL. One is that multiple learning tasks share one low-dimension feature space, which is natural for many applications. For example, [1] proposes a method to learn the shared low-dimension feature space. [11] assumes the cluster structure shared among tasks, which is shown to be equivalent with the low-dimension feature space assumption [27]. In this paper, we refer to the structure induced by the low-dimension space assumption as the low-rank structure. Sparsity is another important structure in machine learning and is introduced to MTL, such as group sparse structure [14, 19] and hierarchical sparse structure [13]. Being sparse means having good properties in theoretical statistics, such as having good consistency

when the sparsity assumption holds [14], as well as having efficiency of prediction and an explicitly interpretable model in practice [2].

Although both the low-rank structure and the sparse structure are explored in previous MTL research, they are seldom considered together in multi-task learning. However, we argue that they should be considered jointly for the following reasons:

- Many MTL problems have both the low-rank structure and the sparse structure. The low-rank structure is usually introduced by the cluster relationship in tasks. When each cluster of tasks shares their own features, the parameter matrix is a block matrix with many blocks of zero matrices, making the parameter matrix both low-rank and sparse.
- The two assumptions are helpful to each other with regard to learning meaningful models for multiple tasks. The low-rank assumption aims to find a small number of factors that govern multiple task learning. The sparsity assumption aims to find the most concise models by dropping irrelevant features. Finding similar tasks with low-rank structure could help capture the relationship among features, which is helpful for learning sparse representations. At the same time, the sparsity requirement may help identify more salient features, which will then lead to better low-rank structures.

In this paper, we exploit both low-rank and sparse structures for MTL. We derive the convex formulation, which regularizes the parameter matrix by both the trace norm and the ℓ_1-norm. Although the problem is convex, it is not easy to solve due to the non-smoothness of the two norms and their different properties [2, 20]. To the best of our knowledge, the standard subgradient method can be used for solving this problem but it can only achieve a rate of convergence $O(1/\sqrt{k})$. To improve the performance of the algorithm, we propose a method combining the Moreau approximation [15] and an accelerated proximal method to optimize the objective. We prove that our method can optimize the objective with a rate of convergence $O(1/k)$, which is much faster than the standard subgradient method.

The contributions of this paper include:

- To the best of our knowledge, we are the first to model both low-rank and sparse structures in multi-task learning.
- We propose a novel optimization algorithm that has the guaranteed rate of convergence $O(1/k)$, which improves the standard subgradient optimization method with a rate of convergence $O(1/\sqrt{k})$.
- We conduct experiments with synthetic data and several real-world data sets and verify the improvement created by exploiting the low-rank and sparse structure in MTL.

2 Notations and Preliminaries

Here is an introduction to the notations used in this paper. Let $\mathbf{Q}$ be a matrix, $\mathbf{q}^i$ denotes the i-th row, $\mathbf{q}_j$ denotes the j-th column and q_{ij} denotes the (i,j) entry. $\sigma_i(\mathbf{Q})$ denotes i-th largest singular value of matrix $\mathbf{Q}$. We define $\mathbb{S}_+^p$ as a set of symmetric positive definite matrices with size $p \times p$. $\mathbf{Tr}(\mathbf{Q})$ denotes the trace of matrix $\mathbf{Q}$. $\|\mathbf{Q}\|_F = \sqrt{\sum_i \sum_j q_{ij}^2}$ denotes the Frobenius norm of matrix $\mathbf{Q}$. $\|\mathbf{Q}\|_0 = \sum_i \sum_j I(q_{ij} \neq 0)$ denotes the ℓ_0-norm of matrix $\mathbf{Q}$. $\|\mathbf{Q}\|_1 = \sum_i \sum_j |q_{ij}|$ denotes the ℓ_1-norm. $\|\mathbf{Q}\|_\infty = \max_{i,j} |q_{ij}|$ denotes the ℓ_∞-norm. $rank(\mathbf{Q})$ denotes the rank of matrix $\mathbf{Q}$. $\|\mathbf{Q}\|_* = \mathbf{Tr}(\mathbf{Q}^T\mathbf{Q})^{\frac{1}{2}} = \sum_i \sigma_i(\mathbf{Q})$ indicates the trace norm (also called nuclear norm) of matrix $\mathbf{Q}$.

In the multi-task learning setting, we have T related tasks and aim to learn functions over the feature space $\mathcal{X} \subseteq \mathbb{R}^p$ given the training data $(\mathbf{x}_i^{(t)}, y_i^{(t)})_{i=1}^{n_t}$. where n_t is the number of training samples for task t, $\mathbf{x}_i^{(t)} \in \mathcal{X}$ is the feature vector for the i-th training sample in the t-th task data and $y_i^{(t)} \in \mathcal{Y}$ is the corresponding response. In binary classification problems, $\mathcal{Y} = \{-1, 1\}$ and in regression problems $\mathcal{Y} \subseteq \mathbb{R}$. We denote the function we need to learn as $f(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$. In this paper, we focus on linear predictors, where $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle$. $\mathbf{w}$ is referred to as the weight vector. Under this setting, the problem of learning $f(\mathbf{x})$ converts to learn the weight vector $\mathbf{w}$.

Let $\mathbf{X}_t = [\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, ..., \mathbf{x}_{n_t}^{(t)}]^T \in \mathbb{R}^{p \times n_t}$ denotes the feature matrix containing each feature vector $\mathbf{x}_i^{(t)}$ as a column, $\mathbf{y}_t = [y_1^{(t)}, y_2^{(t)}, ..., y_{n_t}^{(t)}]^T \in \mathbb{R}^{n_t}$ denotes the response vector for task t, and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T] \in \mathbb{R}^{p \times T}$ is defined as the parameter matrix that we need to estimate. The general formulation for multi-task learning can be expressed as:

$$\min_{\mathbf{W}} \quad \sum_t \mathcal{L}(\mathbf{X}_t^T \mathbf{w}_t, \mathbf{y}_t) + \lambda \Omega(\mathbf{W})$$

where $\mathcal{L}(\mathbf{X}_t^T \mathbf{w}_t, \mathbf{y}_t) : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}$ is a loss function with respect to $\mathbf{w}_t$ to penalize the prediction error. For simplicity of notation, we denote the whole loss function $\sum_t \mathcal{L}(\mathbf{X}_t^T \mathbf{w}_t, \mathbf{y}_t)$ as $\mathcal{L}(\mathbf{W})$. For the whole paper, we assume $\mathcal{L}(\mathbf{W})$ is convex and differentiable and has Lipschitz continuous gradient function. $\lambda \Omega(\mathbf{W})$ is the penalty on the parameter matrix $\mathbf{W}$ to encode some structures of $\mathbf{W}$, typically it is a non-smooth norm. For instance, [9] uses $\Omega(\mathbf{W}) = \|\mathbf{W}\|_*$, which is the convex surrogate of rank of $\mathbf{W}$, to enforce the low-rank structure of $\mathbf{W}$.

3 Problem Formulation

In our model, we expect $\mathbf{W}$ to be simultaneously sparse and low-rank. Formally, we propose the following convex problem:

$$\min_{\mathbf{W}} \quad F(\mathbf{W}) \triangleq \mathcal{L}(\mathbf{W}) + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{W}\|_* \tag{1}$$

where λ_1 and λ_2 are the regularization coefficients to balance the strength of the loss and regularization terms. We observe that there is actually a trade-off between the low-rank structure and the sparse structure. To better visualize this, we consider several special cases of our formulation. When $\lambda_1 = 0$, the problem degenerates to a model with only low-rank structure. This degenerated model is the same as the models in [9] that learn the small number of shared features among tasks and get a convex formulation. When $\lambda_2 = 0$ the problem degenerates into model with only sparse structure where each task is independent and regularized by the ℓ_1-norm. Therefore, we can adjust λ_1, λ_2 to balance the sparse and low-rank structures, and take advantages of both properties.

Moreover, we can easily generalize our model to encode prior structure information into the sparse regularizer. For example, we can replace our ℓ_1-norm regularization with the ℓ_1/ℓ_2-norm: $\lambda_1 \sum_i \|\mathbf{w}^i\|_2$, which is first used in group lasso [23]. Then our reduced model (when $\lambda_2 = 0$) is the same as work [19] which uses the ℓ_1/ℓ_2-norm to encode group sparsity. Similarly, we can generalize our model to encode the hierarchical structure of features into our model by replace our ℓ_1-norm with $\sum_i \sum_g \alpha_{i,g} \|\mathbf{w}_g^i\|_2$. This structure was first introduced to MTL in [13], which is also a special case of our generalized model (when $\lambda_2 = 0$). We will show in the next section that all these generalized models can be efficiently solved by our optimization method.

Also note that our model is substantially different with robust multi-task learning models [6, 8]. Because the two works are similar, we use the model in [6] as an example. They decomposes the parameter matrix $\mathbf{W}$ into two parts and use a trace-norm and a ℓ_1-norm on each part. The goal of their model is to distinguish the outlier features or tasks to maintain the robustness of the model. In comparison, our model aims to penalize $\mathbf{W}$ with both trace-norm and ℓ_1-norm simultaneously to have both low-rank and sparse structures of $\mathbf{W}$. Our goal is to learn concise and more interpretable structure of the tasks with the help of sparsity.

The next result shows how we can learn the parameter matrix for MTL by balancing the effects of low-rank structure and sparse structure.

Theorem 1. *We assuming each task has the same number of samples and denote this number as n. Each task's feature matrix $\mathbf{X}_t \in \mathbb{R}^{p \times n}$ has all columns $i.i.d.$ sampled from a p-variate $\mathcal{N}(0, \mathbf{\Sigma})$ distribution. We denote the true parameter matrix as $\mathbf{W}^*$. We assume $\mathbf{W}^*$ to be both low-rank and sparse, that is $rank(\mathbf{W}^*) \leq r$ and $\|\mathbf{W}^*\|_0 \leq q$, where r and q are two positive integers. Response vector of each task is generated as $\mathbf{y}_t = \mathbf{X}_t^T \mathbf{w}_t^* + \mathbf{b}_t$, where $\mathbf{b}_t$ is a noise with all entries $i.i.d$ sampled from $\mathcal{N}(0, \sigma_w^2)$. We denote $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_T]$ as the noise matrix. We consider the case where the loss is square loss, that is $\sum_t \mathcal{L}(\mathbf{X}_t^T \mathbf{w}_t, \mathbf{y}_t) = \frac{1}{N} \|\mathbf{y}_t - \mathbf{X}_t^T \mathbf{w}_t\|_2^2$ where N is the total number of samples and $N = nT$. The minimizer of the objective Eq (1) is denoted as $\hat{\mathbf{W}}$. The error is defined as $\mathbf{\Delta} = \hat{\mathbf{W}} - \mathbf{W}^*$. By choosing $\lambda_1 + \lambda_2 \geq 2 \frac{12\sqrt{p+T}\sigma_w \sqrt{\sigma_{max}(\mathbf{\Sigma})}}{n^{\frac{1}{4}}}$ we can bound*

the Frobenius norm of $\mathbf{\Delta}$ *with probability* $1 - c_1 \exp(-c_2(m_1 + m_2)\sqrt{n})$*, where* c_1 *and* c_2 *are constants.*

$$\|\mathbf{\Delta}\|_F \leq (2\lambda_1\sqrt{2r} + 2\lambda_2\sqrt{q})\frac{9}{\sigma_{min}(\mathbf{\Sigma})}$$

Moreover,

$$\|\mathbf{\Delta}\|_F \leq (\beta\sqrt{2r} + (1-\beta)\sqrt{q})\frac{432\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\mathbf{\Sigma})}}{\sigma_{min}(\mathbf{\Sigma})n^{\frac{1}{4}}} \tag{2}$$

where β *can be arbitrary value in* $[0, 1]$

The proof generalizes the result of low-rank matrix estimation in [16] and can be found in the appendix. From Eq (2) we can observe that the bound is actually a trade-off between the results from the trace norm and the ℓ_1-norm. We consider two extreme cases: when $\sqrt{q} \ll \sqrt{2r}$ we will take $\beta \approx 0$, Eq (2) is dominated by a bound for lasso. When $\sqrt{2r} \ll \sqrt{q}$ we will take $\beta \approx 1$, Eq (2) is dominated by the bound for multi-task in [16]. Therefore, we can adjust $\beta \in [0, 1]$ according to the strength of low-rank and sparse structures to balance the generalization bound induced by both structures, and take advantages of both properties.

4 Optimization Algorithm

In this section, we first introduce the Moreau approximation method to provide an approximation of Eq (1). Next, we show that by solving an appropriate approximation with a rate of convergence $O(1/k^2)$ we can optimize Eq (1) with a rate of convergence $O(1/k)$. We then present the accelerated proximal method to optimize the approximate objective with a rate of convergence $O(1/k^2)$.

4.1 Smoothing Method

The optimization algorithm for convex objective in Eq (1) is non-trivial. This is because both $\|\mathbf{W}\|_1$ and $\|\mathbf{W}\|_*$ are non-smooth. For such an objective, the best optimization method we know of is the subgradient method, which has been proven to achieve a rate of convergence $O(1/\sqrt{k})$ [17] .

To improve the efficiency of the optimization algorithm, we further exploit the structures of the objective. An objective with either a ℓ_1-norm or a trace norm as the only regularizer can be optimized with a rate of convergence $O(1/k^2)$. However, when the two regularizers are combined, the objective is difficult to optimize because the two regularizers are both non-smooth and have different properties. This inspires us to reduce the two regularizers to one regularizer via a smoothing method. Therefore, we use Moreau proximal smoothing [15] on the trace norm regularizer. More formally, we use the Moreau approximation $\Omega_\mu(\mathbf{W})$ to approximate the trace norm $\lambda_2\|\mathbf{W}\|_*$

$$\Omega_\mu(\mathbf{W}) = \min_{\mathbf{M}}(\frac{1}{2\mu}\|\mathbf{W} - \mathbf{M}\|_F^2 + \lambda_2\|\mathbf{M}\|_*), \tag{3}$$

where μ is the smoothing parameter. The Moreau smooth approximation has a good property that even the convex regularizer on $\mathbf{W}$ is non-smooth, its approximation $\Omega_\mu(\mathbf{W})$ is convex and smooth with respect to $\mathbf{W}$. Moreover, the gradient can easily be computed as

$$\nabla\Omega_\mu(\mathbf{W}) = \lambda_2(\mathbf{W} - \mathbf{M}^*(\mathbf{W})) \tag{4}$$

where $\mathbf{M}^*(\mathbf{W}) = \arg\min_M (\frac{1}{2\mu}\|\mathbf{W} - \mathbf{M}\|_F^2 + \lambda_2\|\mathbf{M}\|_*)$. For the trace norm, we can determine the closed-form expression of $\mathbf{M}^*(\mathbf{W})$ using the soft-threshold operation on the singular values of $\mathbf{W}$ [12]. Next, we replace the trace norm with its Moreau approximation in Eq (1) and obtain the approximated objective with only one non-smooth term.

$$\min_{\mathbf{W}} \quad F_\mu(\mathbf{W}) \triangleq \mathcal{L}(\mathbf{W}) + \lambda_1\|\mathbf{W}\|_1 + \Omega_\mu(\mathbf{W}) \tag{5}$$

We can define the smooth component in Eq (5) as $P_\mu(\mathbf{W}) \triangleq \mathcal{L}(\mathbf{W}) + \Omega_\mu(\mathbf{W})$ and the objective $F_\mu(\mathbf{W})$ can be seen as the summation of the smooth term $P_\mu(\mathbf{W})$ and the simple non-smooth ℓ_1-norm regularization term $\lambda_1\|\mathbf{W}\|_1$.

$$\min_{\mathbf{W}} \quad F_\mu(\mathbf{W}) = P_\mu(\mathbf{W}) + \lambda_1\|\mathbf{W}\|_1 \tag{6}$$

One important question about the smoothing method is how precise can the approximation be. We will show below that based on properly selected smoothing parameter μ, we can solve the problem Eq (1) efficiently at arbitrary precision. Before giving the theorem, we first give two lemmas.

Lemma 1. *For any W, we have $|\Omega_\mu(\mathbf{W}) - \lambda_2\|\mathbf{W}\|_*| \leq c\mu$, where c is a constant independent of μ.*

Lemma 2. *For any W, the Lipschitz constant of the gradient function of $\Omega_\mu(\mathbf{W})$ is less than or equal to $1/\mu$.*

The proof of the two lemmas can be found in [15]. The first lemma shows that the precision of the approximation is proportional to the smoothing parameter μ. The second lemma shows that the Lipschitz constant of the gradient function, which is a measure of the smoothness of a function, is proportional to the inverse of μ. The two lemmas show an interesting conflict between the precision and smoothness when using the smooth approximation. This conflict, as we shall show in Theorem 2, leads a rate of convergence $O(1/k)$ by optimize approximation objective Eq (5) with rate $O(1/k^2)$. This theorem is similar with Theorem 3.1 in a very recent work [5].

Theorem 2. *Suppose that $\mathbf{W}^*$ is the minimizer of Eq (5), and that α is a constant, and that L_P is the Lipschitz constant of the gradient function of the smooth component in Eq (5). Suppose that $\mathcal{M}$ is an iterative method for solving Eq (5) which generates a sequence $\{\mathbf{W}_k\}$. We assume that the sequence satisfies the condition:*

$$F_\mu(\mathbf{W}_k) - F_\mu(\mathbf{W}^*) \leq \frac{L_P\alpha}{k^2} \tag{7}$$

Then by solving Eq (5) with proper μ, we can solve Eq (1) with the rate of convergence $O(L/k)$, where L is the Lipschitz constant of the gradient function of the smooth component in Eq (1).

4.2 Approximate Proximal Method

We now present an algorithm to optimize Eq (5), which can converge on global minimum and generate the sequence satisfying Eq (7). According to Theorem 2, the same algorithm can optimize Eq (1) with a rate of convergence $O(L/k)$. Our algorithm is based on FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [4], which is a popular version of accelerated proximal methods (APM) [18]. APM can be adopted when the objective is able to be decomposed into a convex smooth term and a "simple" non-smooth convex term, where "simple" means the minimizer of the summation of the non-smooth term and a quadratic auxiliary term can be easily obtained. The key step of APM is taking the proximal operator after taking the gradient step in each iteration. In Eq (5), based on the gradient step of the smooth component $P_\mu(\mathbf{W})$ in each iteration, we use the proximal operator to deal with the non-smooth ℓ_1-norm component:

$$\begin{aligned}
\mathbf{W}_{i+1} &= prox_{\frac{\lambda_1}{L_i}\|\cdot\|_1}(\mathbf{W}_i - \frac{1}{L_i}\nabla P_\mu(\mathbf{W}_i)) \\
&= \arg\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W} - (\mathbf{W}_i - \frac{1}{L_i}\nabla P_\mu(\mathbf{W}_i))\|_F^2 \\
&\quad + \frac{\lambda_1}{L_i}\|\mathbf{W}\|_1 \\
&= \arg\min_{\mathbf{W}} P_\mu(\mathbf{W}_i) + \nabla P_\mu(\mathbf{W}_i)^T(\mathbf{W} - \mathbf{W}_i) \\
&\quad + \frac{L_i}{2}\|\mathbf{W} - \mathbf{W}_i\|_F^2 + \lambda_1\|\mathbf{W}\|_1 + const
\end{aligned} \tag{8}$$

where $prox_{h(\cdot)}(\mathbf{W}) \triangleq \arg\min_{\mathbf{M}} \frac{1}{2}\|\mathbf{W} - \mathbf{M}\|_F^2 + h(\mathbf{M})$ is called the proximal operator [2]. It actually linearizes the smooth component near the current point and solve the minimization problem when adding the non-smooth component. The quadratic term $\frac{L_i}{2}\|\mathbf{W} - \mathbf{W}_i\|_F^2$ in Eq (8) is called the proximal term, which can enforce the minimizer to be chosen near $\mathbf{W}_i$. The L_i is a parameter that should be no smaller than the Lipschitz constant of the gradient of smooth component L_p and is determined by backtracking [4]. Also note that when the non-smooth component disappears (i.e. $\lambda_1 = 0$), we get a proximal method for optimizing the smooth component $P_\mu(\mathbf{W})$, which is equivalent to the gradient descent. At this point, we should view the proximal gradient method as a generalization of the gradient method to deal with a non-smooth component.

It has been shown in [4] that FISTA can achieve a rate of convergence $O(L_P/k^2)$. Therefore, the efficient computation of the proximal operator is important for attaining this fast rate. Fortunately, by utilizing the separability of Eq (8), we can derive a closed-form solution [2], which performs the soft-threshold operation on each entry of $\mathbf{W}$. Note that the generalized models introduced in the above section can also be efficiently solved by the closed-form solution of the proximal operator of these structured sparsity norms [2]. Therefore, the generalized models can be solved by our method without loss of computational efficiency.

Algorithm 1 APM for the Trace-norm and the ℓ_1-norm

Input: $\mathbf{W}_0$, L_0, $P_\mu(\mathbf{W})$, $\lambda_1, \lambda_2, \mu$
Output: $\mathbf{W}^*$
Initialize: $t_1 = 1$, $\mathbf{S}_1 = \mathbf{W}_0$, $i = 1$
while $\mathbf{W}$ does not converge **do**
  Determine L_i by backtracking [4].
  $\mathbf{W}_i \leftarrow prox_{\frac{\lambda_1}{L_i}\|\cdot\|_1}(\mathbf{S}_i - \frac{1}{L_i}\nabla P_\mu(\mathbf{S}_i))$
  $t_{i+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_i^2})$
  $\mathbf{S}_{i+1} \leftarrow \mathbf{W}_i + \frac{t_i - 1}{t_{i+1}}(\mathbf{W}_i - \mathbf{W}_{i-1})$
  $i \leftarrow i + 1$
end while
$\mathbf{W}^* \leftarrow \mathbf{W}_i$

4.3 Time complexity analysis

Due to the iterative nature of the algorithm, the time complexity depends on two factors, the number of iterations before convergence and the time consumed in a single iteration. We give the theorem that guarantees the rate of convergence $O(1/k)$ and the total computation time as follows (proof can be found in the appendix):

Theorem 3. *Algorithm 1 converges to the global minimum of objective Eq (1) with rate of convergence $O(1/k)$. The total time consumed by Algorithm 1 is $O(Np + m)/k)$, where m is the computation time for the SVD of matrix $\mathbf{W}$.*

5 Related Work

To the best of our knowledge, there are two closely related works that consider both low-rank and sparse structures. In this section, we discuss them in detail and point out their differences from our work. The first work is [25], which proposes a probabilistic latent factor model for MTL and uses the Laplacian distribution as the prior of the latent vectors, which can introduce ℓ_1-norm in MAP. There are two differences compared with our work. First, this model puts a sparsity constraint on the latent factors rather than the parameter matrix. The sparsity of latent factors does not necessarily lead to a sparse parameter matrix. Our model directly puts the ℓ_1-norm on $\mathbf{W}$, which has guaranteed $\mathbf{W}$ will be sparse. Second, they use a matrix factorization model, which is not formulated as a convex problem, while our model is convex and a global optimal can be found.

The second work is quite recent [21]. The authors analyze the problem of matrix completion with both low-rank and sparse structures. The main differences with our work include:

- Their goal is to find the approximate matrix for some given entries, while we focus on exploiting the low-rank structure and sparse structure in the MTL setting where the parameter matrix is learned.
- They directly optimize the regularization with both the trace norm and the ℓ_1-norm, and their optimization method has no guarantee for the rate of convergence. However, we propose a method that guarantees a rate of convergence $O(1/k)$.

6 Experiment

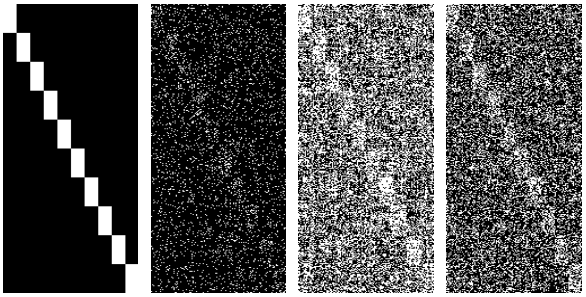
In this section, we present the models for comparison, the results on synthetic data, real data sets used for experiments, the evaluation methods, and the experiment results, as well as some discussions.

6.1 Experiment Setting

Now we introduce the five models compared in our experiment. **LSS** indicates the multi-task learning model proposed in this work. **MTFL** indicates he multi-task feature learning [9] model with square loss. **Lasso** indicates he formulation with square loss and the ℓ_1-norm regularization on the parameter matrix. Note that it can be seen as a summation of the objectives for single tasks, thus it is not a MTL model. **RMTL** [8] indicates the robust multi-task learning model, which is shown to be the state-of-the-art MTL method. This method can simultaneously learn the task relationship and distinguish outliers. **CMTL** [11] indicates a convex model that can learn the cluster structure among multiple tasks. Note that we do not compare the ASO model [7] because ASO has been proven to be mathematically equivalent to CMTL [27].

6.2 Synthetic Data

We first construct a synthetic dataset for experiments. We generate 10 task clusters. Each task cluster has 12 tasks and tasks in one cluster use the same set of 30 features. Then we get the parameter matrix $\mathbf{W} \in \mathbb{R}^{p\times T}$ where $p = 300$ and $T = 120$. For simplicity, we set all non-zero entries in the matrix to have value of 1.0. The parameter matrix is displayed in Figure 1(a). For each task we generate $n_t = 20$ data samples, where the feature vector $\mathbf{x}$ are drawn i.i.d. from $\mathcal{N}(0, \mathbf{I})$ and the response is generated by $y = \mathbf{x}^T\mathbf{w} + \epsilon$ where ϵ is a noise drawn from $\mathcal{N}(0, 5)$. We show the parameter matrix recovered by Lasso, MTFL, LSS in Figure 1(b),1(c), and 1(d), respectively. We can observe that the LSS model can better recover the parameter matrix despite the noise and small number of samples. It is also observed that LSS model can better capture the both the low-rank and sparse structures of parameter matrix.



(a) Ground Truth (b) Lasso (c) MTFL (d) LSS

Figure 1: Parameter matrix recovered by each model

To show the efficiency of our optimization method, we compare our proposed optimization method with some other candidates. Since the model is being proposed for the first time, we do not know any other specially designed optimization methods we only compared to two general optimization methods. Therefore, we compare our method (called LSS-SmoothAPM) with the standard subgradient method [17] (called LSS-subgradient) and the general forward-backward splitting method (called LSS-GFB) recently introduced in [21]. We set these methods with the same initial parameters and all specific parameters for each method (such as step size) are tuned to achieve its best efficiency. We can see the convergence in Figure 2 that our method converges much faster than the other two methods, which is consistent with the theoretical analysis of our method.

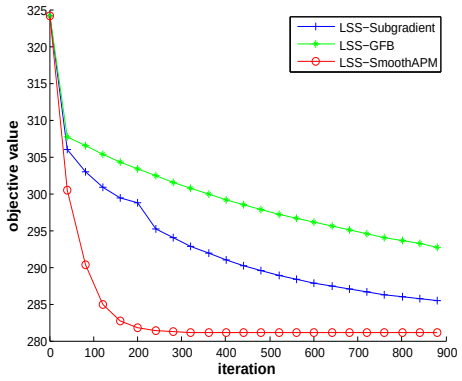


Figure 2: Convergence of LSS-subgradient, LSS-GFB and LSS-SmoothAPM on synthetic data

Table 1: Statistics of the benchmarks data sets

DATA	SAMPLE SIZE	DIMENSION	TASK NUMBER	TYPE
SCHOOL	15,392	27	139	SURVEY
YEAST	2,417	103	14	GENE
SCENE	2,407	294	6	IMAGE
20NEWSGROUP	19,928	62,061	20	TEXT

6.3 Real Data Sets

We also use four real-world data sets for more experimental studies. The datasets include **School data** [26, 3] **Scene data**,**Yeast data**[1] [6] and **20 Newsgroups data**[2] [1]. The School data is a multi-task regression problem and the other three are multi-class classification problems for which we view classifying each class as a task. Table 1 summarizes the statistical information of each data set.

For all benchmark data sets except the 20 newsgroup, we randomly sample 10% from the data sets as the training sets and use the rest 90% as the test sets. The reason for the small sampling ratio is that multi-task learning is well-suited for situations where only a small set of training data exists for each task. For 20 newsgroup, we sample only 5% as a training set and the remaining is used as the test set because it has been shown that 20 newsgroup data set can be predicted well even when there is a relatively small training set [1]. For the regression problem, we report the normalized mean square error (NMSE) and averaged mean square error (AMSE), which were also used in previous work [9]. For the classification problem, we report the average Area Under the Curve (AUC), Macro F1, and Micro F1. The definition of these three metrics can be found in [22]. For all the data sets, we run 15 rounds of experiments and report the mean and variance for all metrics. We tune algorithm parameters via cross-validation.

6.4 Results

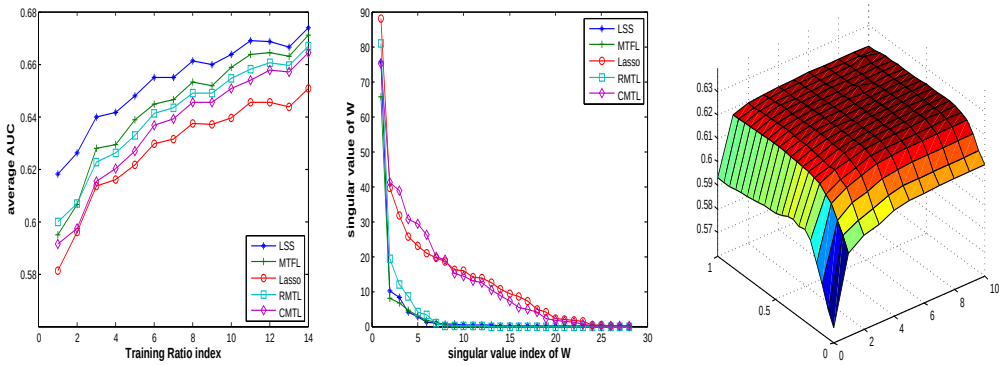
We present the average performance and standard deviation for all five algorithms on the four benchmark data sets in Table 2. From the table, we can reach the following conclusions:

1. LSS outperforms MTFL and Lasso on all data sets. This supports the claim that simultaneously considering low-rank and sparse structures will improve the generalization performance.
2. LSS outperforms RMTL on the gene and text classification data and achieve similar performances on the image classification data. This indicates that considering sparsity in addition to low-rank structure is at least as important as enforcing robustness to for these types of tasks.

[1]Available at `http://www.csie.ntu.edu.tw/~cjlin`
[2]Available at `http://www.ai.mit.edu/~jrennie/20Newsgroups/`

Table 2: Performance comparison of the five competing models in terms of multiple metrics on four data sets.

METRIC	DATA/METHOD	LSS	MTFL	LASSO	RMTL	CMTL
NMSE	SCHOOL	0.8111 ± 0.0161	0.8392 ± 0.0367	0.9088 ± 0.0231	$\mathbf{0.7972 \pm 0.0144}$	0.9139 ± 0.0276
AMSE	SCHOOL	0.2240 ± 0.0049	0.2317 ± 0.0103	0.2510 ± 0.0065	$\mathbf{0.2201 \pm 0.0044}$	0.2524 ± 0.0079
AVERAGE AUC	SCENE	$\mathbf{0.8761 \pm 0.0046}$	0.8703 ± 0.0039	0.8535 ± 0.0177	$\mathbf{0.8768 \pm 0.0021}$	0.8709 ± 0.0061
	YEAST	$\mathbf{0.6260 \pm 0.0076}$	0.6055 ± 0.0110	0.5960 ± 0.0062	0.6069 ± 0.0088	0.6021 ± 0.0103
	20NEWSGROUP	$\mathbf{0.8989 \pm 0.0031}$	0.8891 ± 0.0029	0.8502 ± 0.0042	0.8882 ± 0.0026	0.8723 ± 0.0070
MACRO F1	SCENE	$\mathbf{0.5686 \pm 0.0063}$	0.5554 ± 0.0059	0.5469 ± 0.0135	$\mathbf{0.5645 \pm 0.0016}$	0.5589 ± 0.0103
	YEAST	$\mathbf{0.4190 \pm 0.0049}$	0.4033 ± 0.0071	0.3944 ± 0.0030	0.4096 ± 0.0044	0.4011 ± 0.0043
	20NEWSGROUP	$\mathbf{0.5100 \pm 0.0054}$	0.4830 ± 0.0053	0.4867 ± 0.0076	0.4878 ± 0.0050	0.4813 ± 0.0067
MICRO F1	SCENE	$\mathbf{0.5669 \pm 0.0031}$	0.5523 ± 0.0041	0.5404 ± 0.0137	$\mathbf{0.5661 \pm 0.0014}$	0.5579 ± 0.0107
	YEAST	$\mathbf{0.4640 \pm 0.0048}$	0.4451 ± 0.0071	0.4406 ± 0.0057	0.4556 ± 0.0043	0.4628 ± 0.0041
	20NEWSGROUP	$\mathbf{0.5169 \pm 0.0057}$	0.4930 ± 0.0061	0.4882 ± 0.0076	0.4980 ± 0.0061	0.4891 ± 0.0070



(a) (b) (c)

Figure 3: (a)Performance comparison of MTL models with different training ratio. (b)The ranked singular values of the learned parameter matrix for each MTL algorithm. (c)Model performance against parameters λ_1 and λ_2. The x-axis (left) represents λ_1 and the y-axis (right) represents λ_2 and the z-axis (vertical) is average AUC score.

We also study the effect of the training ratio on model performance. We vary the training ratio from 5% to 70% with a increment of 5% and report the average evaluation results based on 15 runs of experiments. Due to the limited space, we only report the results on the Yeast data set on the AUC metric here. However, the result for other data sets and metrics are similar. The results can be seen in Figure 3(a). From the figure, we find that: (1) For all models, the overall performance improves as the training ratio is increased. This is reasonable since more data is available for model training. (2) The difference between models becomes smaller as the training ratio increase. This indicates that multi-task learning is desirable especially when we lack training data. (3) LSS achieves the best results at almost all settings. This verifies the ability and benefit of LSS when simultaneously learning low-rank and sparse structures from multiple tasks.

6.5 Discussions

In this subsection, we investigate if our proposed LSS algorithm is able to learn low-rank and sparse structures from multiple tasks. Figure 3(b) plots the ranked singular values of each parameter matrix learned by the corresponding algorithm in School data. Note that due to limited space we cannot plot all figures here, however we report that the phenomenons are similar. We can observe that: (1) Most singular values learned by LSS, MTFL and RMTL are close to zero. While singular values learned by Lasso and CMTL are relatively larger than other models. This is because LSS, MTFL and RMTL have the trace norm regularization to encourage low trace, while Lasso and CMTL do not have such a regularization. (2) MTFL has the smallest singular values. LSS also obtains a low trace result, but is slightly larger than MTFL. This is easy to understand since LSS is making a trade-off between low trace and sparsity. (3) low trace could be a good approximation for low-rank since most of the singular values are close to zeros with the trace norm regularization.

Table 3: Sparsity comparisons among MTL algorithms on four benchmark data sets.

DATA/METHOD	LSS	MTFL	LASSO	RMTL	CMTL
SCHOOL	74.7%	100.0%	33.0%	100.0%	100.0%
SCENE	72.6%	100.0%	70.5%	100.0%	100.0%
YEAST	83.4%	100.0%	26.3%	100.0%	100.0%
20NEWSGROUP	1.98%	100.0%	1.97%	100.0%	100.0%

In Table 3, we compare the sparsity of the learned parameter matrices. This comparison is conducted among different MTL algorithms on all four data sets. The values in Table 3 indicate the percentage of nonzero entries in the learned parameter matrices. The smaller the value, the sparser the learned parameter matrix. We can find that MTFL, RMTL and CMTL have no zero entries in their learned parameter matrices, which is reasonable because they do not learn sparse structures. LSS is able to learn sparse structures but the learned parameter matrix is not as sparse as that of Lasso. This is reasonable since LSS balances the structures of low-rank and sparsity. Due to the sparsity nature of text data, LSS and Lasso obtain extremely sparse parameter matrices (less than 2% of non-zero entries) on the 20 newsgroup data set. It is worth noting that other methods cannot achieve sparse models even on text data.

In order to study the influence of learned structures on model performance, we vary the parameters λ_1 and λ_2 and show the result (average AUC) of the Yeast data. We can not report results on other data sets or other metrics due to limited space, however we find that the other data sets show similar results. We change λ_1 from 0 to 1 with step length of 0.05 and λ_2 from 0 to 10 with step length of 1.0. In Figure 3(c), we plot the AUC evaluation scores over the whole parameter space. We can see from Figure 3(c) that only considering low-rank structure or sparse structure can not lead to an optimal model. The best performance is obtained when $\lambda_1 = 0.3$ and $\lambda_2 = 3.0$, i.e., the learning of low-rank and sparse structures is considered simultaneously and is well-balanced.

7 Conclusion and Future Work

In multi-task learning (MTL), both low-rank structure and sparse structure are important but are quite different in nature. We proposed a MTL formulation to learn both low-rank and sparse structures. In order to have an efficient solution, we propose a method for combining the Moreau approximation and APM, which achieves a rate of convergence $O(1/k)$. The experiments on synthetic data and four benchmark data sets demonstrate the effectiveness of our model.

References

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

[2] F. Bach. Optimization with Sparsity-Inducing Penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2011.

[3] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *JMLR*, 4:83–99, 2003.

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

[5] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[6] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *SIGKDD*, pages 1179–1188. ACM, 2010.

[7] J. Chen, L. Tang, J. Liu, and J. Ye. A convex formulation for learning shared structures from multiple tasks. In *ICML*, pages 137–144. ACM, 2009.

[8] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *SIGKDD*, pages 42–50. ACM, 2011.

[9] A. Evgeniou and M. Pontil. Multi-task feature learning. In *NIPS*, volume 19, page 41. MIT Press, 2007.

[10] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *SIGKDD*, pages 109–117. ACM, 2004.

[11] L. Jacob, F. Bach, and J. Vert. Clustered multi-task learning: A convex formulation. *Arxiv preprint arXiv:0809.2085*, 2008.

[12] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, pages 457–464. ACM, 2009.

[13] S. Kim and E. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. *Arxiv preprint arXiv:0909.1373*, 2009.

[14] K. Lounici, M. Pontil, A. Tsybakov, and S. Van De Geer. Taking advantage of sparsity in multi-task learning. *Arxiv preprint arXiv:0903.1468*, 2009.

[15] J. Moreau. Proximitéet dualité dans un espace hilbertien.(french). *Bull. Soc. Math. France*, 93:273–299, 1965.

[16] S. Negahban and M. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

[17] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2003.

[18] Y. Nesterov. Gradient methods for minimizing composite functions. *preprint*, 2007.

[19] G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection for grouped classification. *Department of Statistics, University of California, Berkeley, Tech. Rep*, 743, 2007.

[20] T. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465, 2010.

[21] E. Richard, P. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *ICML*, 2012.

[22] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, ICML, pages 412–420, 1997.

[23] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[24] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. *NIPS*, 18:1585, 2006.

[25] J. Zhang, Z. Ghahramani, and Y. Yang. Flexible latent variable models for multi-task learning. *Machine Learning*, 73(3):221–242, 2008.

[26] Y. Zhang and D. Yeung. A convex formulation for learning task relationships in multi-task learning. In *UAI*, pages 733–742, 2010.

[27] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, 2011.

8 Appendix

8.1 Proof of Theorem 1

We give a lemma which is the multi-task version of Lemma 2 in [16]

Lemma 3. *Consider $\{\mathbf{X}_1, ..., \mathbf{X}_T\}$, where $\mathbf{X}_t \in \mathbb{R}^{p\times n}(t = 1, 2, ..., T)$ is a random matrix with $i.i.d.$ columns sampled from a p-variate $\mathcal{N}(0, \mathbf{\Sigma})$ distribution. Then for $n \geq 2p$ we have*

$$P(\min_t\{\sigma_{min}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T)\} \geq \frac{\sigma_{min}(\mathbf{\Sigma})}{9}, \max_t\{\sigma_{max}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T)\} \leq 9\sigma_{max}(\mathbf{\Sigma})) > 1 - 4T\exp(-\frac{n}{2}) \tag{9}$$

Proof. First, because all matrices are $i.i.d.$, so the probability in Eq (9) is equal to

$$\begin{aligned}&\prod_t P(\sigma_{min}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T) \geq \frac{\sigma_{min}(\mathbf{\Sigma})}{9}, \sigma_{max}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T) \leq 9\sigma_{max}(\mathbf{\Sigma}))\\ =&P(\sigma_{min}(\frac{1}{n}\mathbf{X}_1\mathbf{X}_1^T) \geq \frac{\sigma_{min}(\mathbf{\Sigma})}{9}, \sigma_{max}(\frac{1}{n}\mathbf{X}_1\mathbf{X}_1^T) \leq 9\sigma_{max}(\mathbf{\Sigma}))^T\end{aligned} \tag{10}$$

According to Lemma 2 in [16] we have

$$P(\sigma_{min}(\frac{1}{n}\mathbf{X}_1\mathbf{X}_1^T) \geq \frac{\sigma_{min}(\mathbf{\Sigma})}{9}, \sigma_{max}(\frac{1}{n}\mathbf{X}_1\mathbf{X}_1^T) \leq 9\sigma_{max}(\mathbf{\Sigma})) \geq 1 - 4\exp(-\frac{n}{2}) \tag{11}$$

Combine Eq (10),Eq (11) and the inequality $(1-x)^a > 1-ax$ $(when\ x > 0, a > 1)$, we can obtain

$$\begin{aligned}
&P(\min_t\{\sigma_{min}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T)\} \geq \frac{\sigma_{min}(\mathbf{\Sigma})}{9}, \max_t\{\sigma_{max}(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^T)\} \leq 9\sigma_{max}(\mathbf{\Sigma})) \\
=&(1 - 4\exp(-\frac{n}{2}))^T \\
\geq&1 - 4T\exp(-\frac{n}{2})
\end{aligned}$$

□

We now give a lemma which is the multi-task version of Lemma 3 in [16]

Lemma 4. *Let* $\mathbf{Z} = [\mathbf{X}_1\mathbf{b}_1, ..., \mathbf{X}_T\mathbf{b}_T] \in \mathbb{R}^{p\times T}$*, then there exist constants* $c_i > 0$ *such that*

$$P(\frac{\|\mathbf{Z}\|_2}{N} \geq \frac{12\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\mathbf{\Sigma})}}{n^{\frac{1}{4}}}) \leq c_1 exp(-c_2(p+T)\sqrt{n}) \tag{12}$$

Proof. Let $S^{m-1} = \{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_2 = 1\}$ denote the Euclidean sphere in m-dimension space. The norm of $\mathbf{Z}$ has the variational representation

$$\frac{\|\mathbf{Z}\|_2}{N} = \frac{1}{N}\sup_{\mathbf{u}\in S^{p-1}}\sup_{\mathbf{v}\in S^{T-1}}\mathbf{u}^T\mathbf{Z}\mathbf{v}$$

Now, following the similar proof of Lemma 3 in [16], we have

$$P(\frac{\|\mathbf{Z}\|_2}{N} \geq 4\delta) \leq 8^{p+T}\max_{\mathbf{u}\in S^{p-1},\mathbf{v}\in S^{T-1}} P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} \geq \delta) \tag{13}$$

Now we should bound quantity $\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N}$ given fixed value $\mathbf{u}, \mathbf{v}$. We first rewritten the quantity as

$$\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} = \frac{1}{N}\sum_t(\mathbf{u}^T\mathbf{X}_t)(\mathbf{b}_t v_t)$$

Recall that (b_{ti})s are $i.i.d.$ sampled from $\mathcal{N}(0, \sigma_w^2)$ distribution. Therefore, According to the property of Gaussian distribution, we can get the distribution of $\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N}$ conditioned on the random matrices $\mathbf{X}_1, ..., \mathbf{X}_T$ as

$$\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} \sim \mathcal{N}(0, \frac{\sigma_w^2}{N^2}\sum_t v_t^2\|\mathbf{u}^T\mathbf{X}_t\|_2^2)$$

We define the variance of Gaussian distribution above as α^2. We have

$$\alpha^2 = \frac{\sigma_w^2}{N^2}\sum_t v_t^2\|\mathbf{u}^T\mathbf{X}_t\|_2^2 \leq \frac{\sigma_w^2}{T^2n}\sum_t v_t^2\sigma_{max}(\frac{\mathbf{X}_t\mathbf{X}_t^T}{n})$$

According to Lemma 3, with probability no smaller than $(1-4T\exp(-\frac{n}{2}))$ we have $\sigma_{max}(\frac{\mathbf{X}_t\mathbf{X}_t^T}{n}) \leq 9\sigma_{max}(\mathbf{\Sigma})$. That is

$$\frac{\sigma_w^2}{T^2n}\sum_t v_t^2\sigma_{max}(\frac{\mathbf{X}_t\mathbf{X}_t^T}{n}) \leq \frac{\sigma_w^2}{T^2n}\sum_t v_t^2 9\sigma_{max}(\mathbf{\Sigma}) = \frac{9\sigma_w^2}{T^2n}\sigma_{max}(\mathbf{\Sigma}) \tag{14}$$

We define the event $\{\alpha^2 \leq \frac{9\sigma_w^2}{T^2n}\sigma_{max}(\mathbf{\Sigma})\}$ as E. Conditioning on event E and its complement E^c, we can bound $P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} > \delta)$ as

$$P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} > \delta) \leq P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} > \delta \mid E) + P(E^c) \tag{15}$$

Recall the tail bound for Gaussian distribution: for a random variable x drawn from $\mathcal{N}(0, \sigma^2)$, the probability that $|x| > t$ can be bounded as $P(|x| > t) < exp(-t^2/2\sigma^2)$ when $t > \sigma$. We use the tail bound on $P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} > \delta \mid E)$. Also note that we have $P(E^c) = 1 - P(E) \leq 4T\exp(-\frac{n}{2})$. Combining the results above, we obtain

$$P(\frac{\mathbf{u}^T\mathbf{Z}\mathbf{v}}{N} > \delta \mid E) + P(E^c) \leq \exp(-\frac{\delta^2}{2\frac{9\sigma_w^2}{T^2n}\sigma_{max}(\mathbf{\Sigma})}) + 4T\exp(-\frac{n}{2}) \tag{16}$$

Combining result of Eq (16), Eq (15) and Eq (13), we get

$$P(\frac{\|\mathbf{Z}\|_2}{N} \geq 4\delta) \leq 8^{p+T}\{\exp(-\frac{\delta^2}{2\frac{9\sigma_w^2}{T^2n}\sigma_{max}(\mathbf{\Sigma})}) + 4T\exp(-\frac{n}{2})\} \tag{17}$$

by setting $\delta^2 = (p+T)\frac{9\sigma_w^2}{T^2\sqrt{n}}\sigma_{max}(\mathbf{\Sigma})$. It satisfies the condition $\delta > \alpha$. From Eq (17) we can get the probability now:

$$P(\frac{\|\mathbf{Z}\|_2}{N} \geq 4\delta) \leq 8^{p+T}\{\exp(-\frac{(p+T)\sqrt{n}}{2}) + 4T\exp(-\frac{n}{2})\}$$

This probability vanishes soon as $n > 16(p+T)$. □

Lemma 5. *We denote the $\mathbf{W}^*$ as the real parameter matrix, $\hat{\mathbf{W}}$ as the minimizer of the objective and $\mathbf{\Delta} = \hat{\mathbf{W}} - \mathbf{W}^*$ as the difference between the two matrices. Then we have*

$$\frac{1}{N}\sum_t \|\mathbf{X}_t^T\mathbf{\Delta}_t\|_2^2 \leq \frac{2}{N}\sum_t (\mathbf{X}_t\mathbf{b}_t)^T\mathbf{\Delta}_t + \lambda_1(\|\mathbf{W}^*\|_* - \|\hat{\mathbf{W}}\|_*) + \lambda_2(\|\mathbf{W}^*\|_1 - \|\hat{\mathbf{W}}\|_1) \tag{18}$$

Proof. According to the definition of minimizer of objective, we have

$$\frac{1}{N}\sum_t \|\mathbf{y}_t - \mathbf{X}_t^T\hat{\mathbf{w}}_t\|_2^2 + \lambda_1\|\hat{\mathbf{W}}\|_* + \lambda_2\|\hat{\mathbf{W}}\|_1 \leq \frac{1}{N}\sum_t \|\mathbf{y}_t - \mathbf{X}_t^T\mathbf{w}_t^*\|_2^2 + \lambda_1\|\mathbf{W}^*\|_* + \lambda_2\|\mathbf{W}^*\|_1$$

Using some algebra we have

$$\frac{1}{N}\sum_t (\|\mathbf{X}_t^T\mathbf{\Delta}_t\|_2^2 + 2(\mathbf{X}_t^T\mathbf{w}_t^* - \mathbf{y}_t)^T(\mathbf{X}_t^T\mathbf{\Delta}_t)) \leq \lambda_1(\|\mathbf{W}^*\|_* - \|\hat{\mathbf{W}}\|_*) + \lambda_2(\|\mathbf{W}^*\|_1 - \|\hat{\mathbf{W}}\|_1)$$

Recall that $\mathbf{y}_t = \mathbf{X}_t^T\mathbf{w}_t^* + \mathbf{b}_t$. We now can obtain

$$\frac{1}{N}\sum_t (\|\mathbf{X}_t^T\mathbf{\Delta}_t\|_2^2 - 2\mathbf{b}_t^T(\mathbf{X}_t^T\mathbf{\Delta}_t)) \leq \lambda_1(\|\mathbf{W}^*\|_* - \|\hat{\mathbf{W}}\|_*) + \lambda_2(\|\mathbf{W}^*\|_1 - \|\hat{\mathbf{W}}\|_1)$$

Move the second term from left side to the right side of the inequality we get the result in Lemma 5. □

Lemma 6. *Define $\mathbf{U} \in \mathbb{R}^{p\times p}$ and $\mathbf{V} \in \mathbb{R}^{T\times T}$ as the matrices consisting of the left and right singular vectors of $\mathbf{W}^*$, respectively. Then there exists a matrix decomposition $\mathbf{\Delta} = \mathbf{\Delta}_L^{'} + \mathbf{\Delta}_L^{''}$ of the error $\mathbf{\Delta}$ such that:*

- *(a) the matrix $\mathbf{\Delta}_L^{'}$ satisfies the constraint $rank(\mathbf{\Delta}_L^{'}) \leq 2r$;*
- *(b) the difference of trace norm between $\hat{\mathbf{W}}$ and $\mathbf{W}^*$ can be bounded as $\|\mathbf{W}^*\|_* - \|\hat{\mathbf{W}}\|_* \leq \|\mathbf{\Delta}_L^{'}\|_* - \|\mathbf{\Delta}_L^{''}\|_*$.*
- *(c) $\|\mathbf{\Delta}_L^{'}\|_F \leq \|\mathbf{\Delta}\|_F$*

Proof. This Lemma can be seen as the special case of Lemma 1 in [16] when $rank(\mathbf{W}^*) \leq r$. For error matrix $\mathbf{\Delta}$, we consider

$$\mathbf{\Gamma} \triangleq \mathbf{U}\mathbf{\Delta}\mathbf{V}^T = \begin{pmatrix} \mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12} \\ \mathbf{\Gamma}_{21} & \mathbf{\Gamma}_{22} \end{pmatrix}$$

where $\mathbf{\Gamma}_{11} \in \mathbb{R}^{r\times r}$ and $\mathbf{\Gamma}_{22} \in \mathbb{R}^{(p-r)\times(T-r)}$. Define matrix $\mathbf{\Delta}_L^{'}$ and $\mathbf{\Delta}_L^{''}$ as

$$\mathbf{\Delta}_L^{''} \triangleq \mathbf{U}\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{\Gamma}_{22} \end{pmatrix}\mathbf{V}^T \quad and \quad \mathbf{\Delta}_L^{'} \triangleq \mathbf{\Delta} - \mathbf{\Delta}_L^{''} \tag{19}$$

We can bound $rank(\Delta_L^{'})$ by

$$rank(\mathbf{\Delta}_L^{'}) = rank\begin{pmatrix} \mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12} \\ \mathbf{\Gamma}_{21} & 0 \end{pmatrix} \leq rank\begin{pmatrix} \mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12} \\ 0 & 0 \end{pmatrix} + rank\begin{pmatrix} \mathbf{\Gamma}_{11} & 0 \\ \mathbf{\Gamma}_{21} & 0 \end{pmatrix} \leq 2r$$

This is the proof of (a) of Lemma 6. Also note that the trace norm satisfies the decomposition below

$$\|\mathbf{W}^* + \mathbf{\Delta}_L^{''}\|_* = \|\mathbf{W}^*\|_* + \|\mathbf{\Delta}_L^{''}\|_* \tag{20}$$

Now, with Eq (19), Eq (20) and triangle inequality, we have

$$\begin{aligned} \|\hat{\mathbf{W}}\|_* =& \|\mathbf{W}^* + \mathbf{\Delta}_L^{''} + \mathbf{\Delta}_L^{'}\|_* \\ \geq& \|\mathbf{W}^* + \mathbf{\Delta}_L^{''}\|_* - \|\mathbf{\Delta}_L^{'}\|_* \\ =& \|\mathbf{W}^*\|_* + \|\mathbf{\Delta}_L^{''}\|_* - \|\mathbf{\Delta}_L^{'}\|_* \end{aligned}$$

And, as a result, we can have

$$\|\mathbf{W}^*\|_* - \|\hat{\mathbf{W}}\|_* \leq \|\mathbf{W}^*\|_* - \|\mathbf{W}^*\|_* - \|\mathbf{\Delta}_L^{''}\|_* + \|\mathbf{\Delta}_L^{'}\|_* = \|\mathbf{\Delta}_L^{'}\|_* - \|\mathbf{\Delta}_L^{''}\|_*$$

Now we establish (b) in Lemma 6. We have

$$\|\mathbf{\Delta}_L^{'}\|_F = \|\begin{pmatrix} \mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12} \\ \mathbf{\Gamma}_{21} & 0 \end{pmatrix}\|_F \leq \|\mathbf{\Gamma}\|_F = \|\mathbf{\Delta}\|_F$$

Then we finish the proof of (c) in Lemma 6. □

Lemma 7. *Define the nonzero indicator matrix of $\mathbf{W}^*$ as $\mathbf{\Theta}$ and $\mathbf{\Theta}^{\perp}\{0,1\}^{p\times T}$ as its complementary. That is $\theta_{ij} = I(w_{ij} \neq 0)$ and $\theta^{\perp} = I(w_{ij} = 0)$. Define $\mathbf{\Delta}_S^{'} = \mathbf{\Theta} \circ \mathbf{\Delta}$ and $\mathbf{\Delta}_S^{''} = \mathbf{\Delta} - \mathbf{\Delta}_S^{'}$, where $A \circ B$ means elementwise product of matrix A and B. Then we have*

- *(a)* $\|\mathbf{\Delta}_S^{'}\|_0 \leq q$
- *(b)* $\|\mathbf{W}^*\|_1 - \|\hat{\mathbf{W}}\|_1 \leq \|\mathbf{\Delta}_S^{'}\|_- \|\mathbf{\Delta}_S^{''}\|_1$
- *(c)* $\|\mathbf{\Delta}_S^{'}\|_F \leq \|\mathbf{\Delta}\|_F$

Proof. According to the definition of $\mathbf{\Delta}_S^{'}$, the set of nonzero entries of $\mathbf{\Delta}_S^{'}$ is a subset of nonzero entries of $\mathbf{W}^*$. Therefore we can get (a) of Lemma 7.

Decomposing $\hat{\mathbf{W}}$ as the summation of $\mathbf{W}^*$ and $\mathbf{\Delta}$ and using the triangle inequality, we can get

$$\begin{aligned} \|\hat{\mathbf{W}}\|_1 =& \|(\mathbf{W}^* + \mathbf{\Delta}_S^{''} + \mathbf{\Delta}_S^{'}\|_1 \\ \geq& \|\mathbf{W}^* + \mathbf{\Delta}_S^{''}\|_1 - \|\mathbf{\Delta}_S^{'}\|_1 \\ =& \|\mathbf{W}^*\|_1 + \|\mathbf{\Delta}_S^{''}\|_1 - \|\mathbf{\Delta}_S^{'}\|_1 \end{aligned} \tag{21}$$

According to Eq (21) we can have

$$\|\mathbf{W}^*\|_1 - \|\hat{\mathbf{W}}\|_1 \leq \|\mathbf{W}^*\|_1 - \|\mathbf{W}^*\|_1 - \|\mathbf{\Delta}_S^{''}\|_1 + \|\mathbf{\Delta}_S^{'}\|_1 = \|\mathbf{\Delta}_S^{'}\|_1 - \|\mathbf{\Delta}_S^{''}\|_1$$

This establishes (b) of Lemma 7 According to the definition of $\mathbf{\Delta}_S^{'}$, the set of nonzero entries of $\mathbf{\Delta}_S^{'}$ is a subset of nonzero entries of $\mathbf{\Delta}^*$. Therefore we prove (c) of Lemma 7. □

We can now give the proof of Theorem 1.

Proof. According to Lemma 5 we have

$$\begin{aligned}&\frac{1}{N}\sum_t \|\mathbf{X}_t^T\boldsymbol{\Delta}_t\|_2^2\\ \leq&\frac{2}{N}\sum_t(\mathbf{X}_t\mathbf{b}_t)^T\boldsymbol{\Delta}_t+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\end{aligned} \tag{22}$$

Recall the definition of $\mathbf{Z}$ in Lemma 4, we have

$$\begin{aligned}&\frac{2}{N}\sum_t(\mathbf{X}_t\mathbf{b}_t)^T\boldsymbol{\Delta}_t+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\\ =&\frac{2}{N}(\mathbf{Z}\circ\boldsymbol{\Delta})+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\\ =&\frac{2\lambda_1}{N(\lambda_1+\lambda_2)}(\mathbf{Z}\circ\boldsymbol{\Delta})+\frac{2\lambda_2}{N(\lambda_1+\lambda_2)}(\mathbf{Z}\circ\boldsymbol{\Delta})+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\\ \leq&\frac{2\lambda_1}{N(\lambda_1+\lambda_2)}\|\mathbf{Z}\|_2\|\boldsymbol{\Delta}\|_*+\frac{2\lambda_2}{N(\lambda_1+\lambda_2)}\|\mathbf{Z}\|_\infty\|\boldsymbol{\Delta}\|_1+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\end{aligned} \tag{23}$$

Where the inequality uses the definition of dual norm. Recall that in Lemma 4 we can bound $\frac{\|\mathbf{Z}\|_2}{N}$ below $\frac{12\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\boldsymbol{\Sigma})}}{n^{\frac{1}{4}}}$ with probability $1-c_1\exp(-c_2(m_1+m_2)\sqrt{n})$. By setting $\lambda_1+\lambda_2\geq 2\frac{12\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\boldsymbol{\Sigma})}}{n^{\frac{1}{4}}}$ we obtain the following inequality with probability $1-c_1\exp(-c_2(m_1+m_2)\sqrt{n})$. Note that $\|\mathbf{Z}\|_\infty\leq\|\mathbf{Z}\|_2$. We can obtain

$$\begin{aligned}&\frac{2\lambda_1}{N(\lambda_1+\lambda_2)}\|\mathbf{Z}\|_2\|\boldsymbol{\Delta}\|_*+\frac{2\lambda_2}{N(\lambda_1+\lambda_2)}\|\mathbf{Z}\|_\infty\|\boldsymbol{\Delta}\|_1+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\\ \leq&\lambda_1\|\boldsymbol{\Delta}\|_*+\lambda_2\|\boldsymbol{\Delta}\|_1+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\end{aligned} \tag{24}$$

By using Lemma 6 and Lemma 7, triangle inequality and Cauchy inequality $\|\mathbf{A}\|_*\leq\sqrt{rank(\mathbf{A})}\|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_1\leq\sqrt{\|\mathbf{A}\|_0}\|\mathbf{A}\|_F$.

$$\begin{aligned}&\lambda_1\|\boldsymbol{\Delta}\|_*+\lambda_2\|\boldsymbol{\Delta}\|_1+\lambda_1(\|\mathbf{W}^*\|_*-\|\hat{\mathbf{W}}\|_*)+\lambda_2(\|\mathbf{W}^*\|_1-\|\hat{\mathbf{W}}\|_1)\\ \leq&\lambda_1\|\boldsymbol{\Delta}\|_*+\lambda_2\|\boldsymbol{\Delta}\|_1+\lambda_1(\|\boldsymbol{\Delta}_L^{'}\|_*-\|\boldsymbol{\Delta}_L^{''}\|_*)+\lambda_2(\|\boldsymbol{\Delta}_S^{'}\|_1-\|\boldsymbol{\Delta}_S^{''}\|_1)\\ \leq&\lambda_1(\|\boldsymbol{\Delta}_L^{'}\|_*+\|\boldsymbol{\Delta}_L^{''}\|_*)+\lambda_2(\|\boldsymbol{\Delta}_S^{'}\|_1+\|\boldsymbol{\Delta}_S^{''}\|_1)+\lambda_1(\|\boldsymbol{\Delta}_L^{'}\|_*-\|\boldsymbol{\Delta}_L^{''}\|_*)+\lambda_2(\|\boldsymbol{\Delta}_S^{'}\|_1-\|\boldsymbol{\Delta}_S^{''}\|_1)\\ =&2\lambda_1\|\boldsymbol{\Delta}_L^{'}\|_*+2\lambda_2\|\boldsymbol{\Delta}_S^{'}\|_1\\ \leq&2\lambda_1\sqrt{2r}\|\boldsymbol{\Delta}_L^{'}\|_F+2\lambda_2\sqrt{q}\|\boldsymbol{\Delta}_S^{'}\|_F\\ \leq&2\lambda_1\sqrt{2r}\|\boldsymbol{\Delta}\|_F+2\lambda_2\sqrt{q}\|\boldsymbol{\Delta}\|_F\\ =&(2\lambda_1\sqrt{2r}+2\lambda_2\sqrt{q})\|\boldsymbol{\Delta}\|_F\end{aligned} \tag{25}$$

Combining Eq (22), Eq (23), Eq (24) and Eq (25) we can get

$$\frac{1}{N}\sum_t\|\mathbf{X}_t^T\boldsymbol{\Delta}_t\|_2^2\leq(2\lambda_1\sqrt{2r}+2\lambda_2\sqrt{q})\|\boldsymbol{\Delta}\|_F$$

And use Lemma 3, with probability $1-4T\exp(-n/2)$ we have

$$\frac{\sigma_{min}(\boldsymbol{\Sigma})}{9}\|\boldsymbol{\Delta}\|_F^2\leq\frac{1}{N}\sum_t\|\mathbf{X}_t^T\boldsymbol{\Delta}_t\|_2^2\leq(2\lambda_1\sqrt{2r}+2\lambda_2\sqrt{q})\|\boldsymbol{\Delta}\|_F \tag{26}$$

Note that this event is implied Eq (24). As a result, when $\lambda_1+\lambda_2=2\frac{12\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\boldsymbol{\Sigma})}}{n^{\frac{1}{4}}}$, we have

$$\|\boldsymbol{\Delta}\|_F\leq(\beta\sqrt{2r}+(1-\beta)\sqrt{q})\frac{432\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\boldsymbol{\Sigma})}}{\sigma_{min}(\boldsymbol{\Sigma})n^{\frac{1}{4}}}$$

This inequality holds with probability $1 - c_1 \exp(-c_2(m_1 + m_2)\sqrt{n})$. Using $\lambda_1 + \lambda_2 \geq 2\frac{12\sqrt{p+T}\sigma_w\sqrt{\sigma_{max}(\mathbf{\Sigma})}}{n^{\frac{1}{4}}}$ in Eq (8.1), and denote $\beta = \frac{\lambda_1}{\lambda_1+\lambda_2}$ we have

$$\|\mathbf{\Delta}\|_F \leq (2\lambda_1\sqrt{2r} + 2\lambda_2\sqrt{q})\frac{9}{\sigma_{min}(\mathbf{\Sigma})}$$

□

8.2 Proof of Theorem 2

Proof. This proof is similar with the proof of Theorem 3.1 in a very recent work [5]. The smooth component $P_\mu(\mathbf{W})$ consists of two parts: the loss function and the smoothed regularizer. The Lipschitz constant of the gradient function of the loss function $\mathcal{L}\mathbf{W}$ is equal to L. And we denote the Lipschitz constant of the gradient function of the smoothed regularizer as L_Ω. Because $P_\mu(\mathbf{W}) = \mathcal{L}(\mathbf{W}) + \Omega_\mu(\mathbf{W})$, we have $L_P = L + L_\Omega$. Denote $\{\mathbf{W}_k\}$ as the sequence generated by method $\mathcal{M}$. $\{\mathbf{W}_k\}$ has the following property

$$F_\mu(\mathbf{W}_k) - F_\mu(W^*) \leq \frac{L_P\alpha}{k^2} = \frac{(L + L_\Omega)\alpha}{k^2} \tag{27}$$

where α is a constant. By Lemma 1 in paper, we can bound the difference between $F(\mathbf{W})$ and $F_\mu(\mathbf{W})$ as

$$\begin{aligned} & F_\mu(\mathbf{W}) - F(\mathbf{W}) \\ =&(\mathcal{L}(\mathbf{W}) + \Omega_\mu(\mathbf{W}) + \lambda_2\|\mathbf{W}\|_1) - (\mathcal{L}(\mathbf{W}) + \lambda_1\|\mathbf{W}\|_* + \lambda_2\|\mathbf{W}\|_1) \\ =&\Omega_\mu(\mathbf{W}) - \lambda_1\|\mathbf{W}\|_* \\ \leq&c\mu \end{aligned} \tag{28}$$

and

$$F_\mu(\mathbf{W}) - F(\mathbf{W}) = \Omega_\mu(\mathbf{W}) - \lambda_1\|\mathbf{W}\|_* \geq 0 \tag{29}$$

Where c is a constant. Combining Eq (27), Eq (28) and Eq (29), we can get

$$F(\mathbf{W}_k) - F(\mathbf{W}^*) \leq (F_\mu(\mathbf{W}_k) + c\mu) - F_\mu(\mathbf{W}^*) \leq \frac{(L + L_\Omega)\alpha}{k^2} + c\mu \tag{30}$$

According to Lemma 2 in paper, we have $L_\Omega = \frac{1}{\mu}$, substitute L_Ω in Eq (30), we get

$$F(\mathbf{W}_k) - F(\mathbf{W}^*) \leq \frac{L\alpha}{k^2} + \frac{\alpha}{\mu k^2} + c\mu \tag{31}$$

We select μ to minimize the right side of Eq (31), according to AM-GM inequality, we have $\mu = \sqrt{\frac{\alpha}{c}}\frac{1}{k}$ and then

$$F(\mathbf{W}_k) - F(\mathbf{W}^*) \leq \frac{L\alpha}{k^2} + 2\frac{\sqrt{\alpha c}}{k} = O(1/k)$$

□

8.3 Proof of Theorem 3

Proof. Because the loss function and the Moreau approximation function are both smooth and have continuous Lipschitz gradient, the smooth part of objective Eq (5) satisfies the condition in [4]. Moreover, the summation of the smooth part and ℓ_1-norm regularizer has a closed-form minimizer by Eq (8). According to [4], Algorithm 1 will converge to the global minimum of approximated objective Eq (5) which satisfies the condition in Theorem 2. According to Theorem 1, Algorithm 1 can converge to the global minimum of the original objective Eq (1) with rate of convergence O(1/k). Summarizing the above points we finish the proof of convergence rate.

As to the time consumed in one iteration, to compute the gradient of the smooth part $P_\mu(\mathbf{W})$, we need to calculate the gradient of the loss function and the gradient of Moreau approximation function. It costs $O(Np)$ and $O(m)$, respectively, where N denotes the total training data points and m is the computation time for SVD of $\mathbf{W}$. The time to consider the ℓ_1-norm is $O(pT)$ which is an order of smaller than the other two terms. Therefore, the time complexity for each iteration is $O(Np + m)$. In conclusion, the total time complexity is $O((Np + m)/k)$ □