

# Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach

Ming Yu, Varun Gupta, and Mladen Kolar

*Booth School of Business, The University of Chicago, Chicago, IL 60637*  
*e-mail:*

[mingyu@chicagobooth.edu](mailto:mingyu@chicagobooth.edu); [varun.gupta@chicagobooth.edu](mailto:varun.gupta@chicagobooth.edu); [mladen.kolar@chicagobooth.edu](mailto:mladen.kolar@chicagobooth.edu)

**Abstract:** We study the problem of recovery of matrices that are simultaneously low rank and row and/or column sparse. Such matrices appear in recent applications in cognitive neuroscience, imaging, computer vision, macroeconomics, and genetics. We propose a GDT (Gradient Descent with hard Thresholding) algorithm to efficiently recover matrices with such structure, by minimizing a bi-convex function over a nonconvex set of constraints. We show linear convergence of the iterates obtained by GDT to a region within statistical error of an optimal solution. As an application of our method, we consider multi-task learning problems and show that the statistical error rate obtained by GDT is near optimal compared to minimax rate. Experiments demonstrate competitive performance and much faster running speed compared to existing methods, on both simulations and real data sets.

**MSC 2010 subject classifications:** Primary 90C26.

**Keywords and phrases:** Nonconvex optimization, low rank and two-way sparse coefficient matrix, gradient descent with hard thresholding, multi-task learning, two-way sparse reduce rank regression.

Received January 2019.

## Contents

1	Introduction . . . . .	414
1.1	Related work . . . . .	416
1.2	Organization of the paper . . . . .	418
2	Gradient descent with hard thresholding . . . . .	418
3	Theoretical result . . . . .	420
3.1	Regularity conditions . . . . .	420
3.2	Main result . . . . .	423
3.3	Proof sketch of Theorem 1 . . . . .	424
4	Application to multi-task learning . . . . .	425
4.1	GDT for multi-task learning . . . . .	425
4.2	Application to multi-task reinforcement learning . . . . .	429
5	Experiment . . . . .	432
5.1	Synthetic datasets . . . . .	432
5.2	Norwegian paper quality dataset . . . . .	436

5.3	Calcium imaging data . . . . .	438
6	Technical proofs . . . . .	439
6.1	Proof of Lemma 2 . . . . .	439
6.2	Proof of Lemma 3 . . . . .	440
6.3	Proof of Lemma 4 . . . . .	444
6.4	Proof of Lemma 5 . . . . .	445
7	Conclusion . . . . .	446
	Acknowledgments . . . . .	446
	References . . . . .	446

## 1. Introduction

Many problems in machine learning, statistics and signal processing can be formulated as optimization problems with a smooth objective and nonconvex constraints. The objective usually measures the fit of a model, parameter, or signal to the data, while the constraints encode structural requirements on the model. Examples of nonconvex constraints include sparsity where the parameter is assumed to have only a few non-zero coordinates [67, 137, 111, 120, 147], group sparsity where the parameter is comprised of several groups only few of which are non-zero [90, 77, 68, 36], and low-rankness where the parameter is believed to be a linear combination of few factors [6, 37, 42, 55, 72]. Common approach to dealing with nonconvex constraints is via convex relaxations, which allow for application of simple optimization algorithms and easy theoretical analysis [2, 30, 51, 29, 80]. From a practical point of view, it has been observed that directly working with a nonconvex optimization problem can lead to both faster and more accurate algorithms [115, 144, 138, 129]. As a result, a body of literature has recently emerged that tries to characterize good performance of these algorithms [13, 143, 57].

In this work, we focus on the following optimization problem

$$\hat{\Theta} \in \arg \min_{\Theta \in \Xi} f(\Theta) \quad (1.1)$$

where  $\Xi \subset \mathbb{R}^{m_1 \times m_2}$  is a nonconvex set comprising of low rank matrices that are also row and/or column sparse,

$$\Xi = \Xi(r, s_1, s_2) = \{\Theta \in \mathbb{R}^{m_1 \times m_2} \mid \text{rank}(\Theta) \leq r, \|\Theta\|_{2,0} \leq s_1, \|\Theta^\top\|_{2,0} \leq s_2\},$$

where  $\|\Theta\|_{2,0} = |\{i \in [m_1] \mid \sum_{j \in [m_2]} \Theta_{ij}^2 \neq 0\}|$  is the number of non-zero rows of  $\Theta$ . Such an optimization problem arises in a number of applications including sparse singular value decomposition and principal component analysis [129, 91, 65], sparse reduced-rank regression [20, 93, 37, 38, 123], and reinforcement learning [26, 116, 83, 131, 113]. Rather than considering convex relaxations of the optimization problem (1.1), we directly work with a nonconvex formulation. Under an appropriate statistical model, the global minimizer  $\hat{\Theta}$  approximates the “true” parameter  $\Theta^*$  with an error level  $\epsilon$ . Since the optimization problem (1.1) is highly nonconvex, our aim is to develop an iterative algorithm

that, with appropriate initialization, converges linearly to a stationary point  $\hat{\Theta}$  that is within  $c \cdot \epsilon$  distance of  $\hat{\Theta}$ . In order to develop a computationally efficient algorithm, we reparametrize the  $m_1 \times m_2$  matrix variable  $\Theta$  as  $UV^\top$  with  $U \in \mathbb{R}^{m_1 \times r}$  and  $V \in \mathbb{R}^{m_2 \times r}$ , and optimize over  $U$  and  $V$ . That is, we consider (with some abuse of notation) the following optimization problem

$$(\hat{U}, \hat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V), \quad (1.2)$$

where

$$\mathcal{U} = \mathcal{U}(s_1) = \{U \in \mathbb{R}^{m_1 \times r} \mid \|U\|_{2,0} \leq s_1\}$$

and

$$\mathcal{V} = \mathcal{V}(s_2) = \{V \in \mathbb{R}^{m_2 \times r} \mid \|V\|_{2,0} \leq s_2\}.$$

Such a reparametrization automatically enforces the low rank structure and will allow us to develop an algorithm with low computational cost per iteration. Note that even though  $\hat{U}$  and  $\hat{V}$  are only unique up to scaling and a rotation by an orthogonal matrix,  $\hat{\Theta} = \hat{U}\hat{V}^\top$  is usually unique.

We make several contributions in this paper. First, we develop an efficient algorithm for minimizing (1.2), which uses projected gradient descent on a non-convex set in each iteration. Under conditions on the function  $f(\Theta)$  that are common in the high-dimensional literature, we establish linear convergence of the iterates to a statistically relevant solution. In particular, we require that the function  $f(\Theta)$  satisfies restricted strong convexity (RSC) and restricted strong smoothness (RSS), conditions that are given in Condition **(RSC/RSS)** below. Compared to the existing work for optimization over low rank matrices with (alternating) gradient descent, we need to study a projection onto a nonconvex set in each iteration, which in our case is a hard-thresholding operation, that requires delicate analysis and novel theory. Our second contribution, is in the domain of multi-task learning. Multi-task learning is a widely used learning framework where similar tasks are considered jointly for the purpose of improving performance compared to learning the tasks separately [32]. We study the setting where the number of input variables and the number of tasks can be much larger than the sample size (see [93] and references there in). Our focus is on simultaneous variable selection and dimensionality reduction. We want to identify which variables are relevant predictor variables for different tasks and at the same time we want to combine the relevant predictor variables into fewer features that can be explained as latent factors that drive the variation in the multiple responses. We provide a new algorithm for this problem and improve the theoretical results established in [93]. In particular, our algorithm does not require a new independent sample in each iteration and allows for non-Gaussian errors, while at the same time achieves nearly optimal error rate compared to the information theoretic minimax lower bound for the problem. Moreover, our prediction error is much better than the error bound proposed in [20], and matches the error bound in [111]. However, all of the existing algorithms are slow and cannot scale to high dimensions. Finally, our third contribution is in the area of reinforcement learning. We study the Multi-task Reinforcement

Learning (MTRL) problem via value function approximation. In MTRL the decision maker needs to solve a sequence of Markov Decision Processes (MDPs). A common approach to Reinforcement Learning when the state space is large is to approximate the value function of linear basis functions (linear in some appropriate feature representation of the states) with sparse support. Thus, it is natural to assume the resulting coefficient matrix is low rank and row sparse. Our proposed algorithm can be applied to the regression step of any MTRL algorithm (we chose Fitted  $Q$ -iteration (FQI) for presentation purposes) to solve for the optimal policies for MDPs. Compared to [26] which uses convex relaxation, our algorithm is much more efficient in high dimensions.

### 1.1. Related work

Our work contributes to several different areas, and thus is naturally related to many existing works. We provide a brief overview of the related literature and describe how it is related to our contributions. For the sake of brevity, we do not provide an extensive review of the existing literature.

**Low-rank matrix recovery** A large body of literature exists on recovery of low-rank matrices as they arise in a wide variety of applications throughout science and engineering, ranging from quantum tomography to signal processing and machine learning [1, 89, 114, 45]. Recovery of a low-rank matrix can be formulated as the following optimization problem

$$\widehat{\Theta} \in \arg \min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) \quad \text{subject to } \text{rank}(\Theta) \leq r, \quad (1.3)$$

where the objective function  $f : \mathbb{R}^{m_1 \times m_2} \mapsto \mathbb{R}$  is convex and smooth. The problem (1.3) is highly nonconvex and NP-hard in general [51, 52]. A lot of the progress in the literature has focused on convex relaxations where one replaces the rank constraint using the nuclear norm. See, for example, [29, 30, 28, 106, 23, 105, 55, 33, 67, 109, 80, 61, 98, 36, 132, 99, 2, 107, 39, 40, 41, 65, 25, 135, 147, 128] and references therein. However, developing efficient algorithms for solving these convex relaxations is challenging in regimes with large  $m_1$  and  $m_2$  [66]. A practical approach, widely used in large scale applications such as recommendation systems or collaborative filtering [117, 81, 54, 150] relies on solving a nonconvex optimization problem where the decision variable  $\Theta$  is factored as  $UV^\top$ , usually referred to as the Burer-Monteiro type decomposition [21, 22]. A stationary point of this nonconvex problem is usually found via a block coordinate descent-type algorithm, such as alternating minimization or (alternating) gradient descent. Unlike for the convex relaxation approaches, the theoretical understanding of these nonconvex optimization procedures has been developed only recently [75, 76, 72, 62, 64, 63, 115, 144, 43, 97, 49, 145, 17, 16, 119, 42, 148, 35, 149, 53, 88, 86, 94, 58]. Compared to the classical nonconvex optimization theory, which only shows a sublinear convergence to a local optima, the focus of the recent literature is on establishing linear rates of

convergence or characterizing that the objective does not have spurious local minima. In addition to the methods that work on the factorized form, [70, 84, 71, 13, 46] consider projected gradient-type methods which optimize over the matrix variable  $\Theta \in \mathbb{R}^{m_1 \times m_2}$ . These methods involve calculating the top  $r$  singular vectors of an  $m_1 \times m_2$  matrix at each iteration. When  $r$  is much smaller than  $m_1$  and  $m_2$ , they incur much higher computational cost per iteration than the methods that optimize over  $U \in \mathbb{R}^{m_1 \times r}$  and  $V \in \mathbb{R}^{m_2 \times r}$ .

Our work contributes to this body of literature by studying gradient descent with a projection step on a non-convex set, which requires hard-thresholding. Hard-thresholding in this context has not been considered before. Theoretically we need a new argument to establish linear convergence to a statistically relevant point. [42] considered projected gradient descent in a symmetric and positive semidefinite setting with a projection on a convex set. Our work is most closely related to [144], which used the notion of inexact first order oracle to establish their results, but did not consider the hard-thresholding step.

**Structured low-rank matrices** Low-rank matrices with additional structure also commonly arise in different problems ranging from sparse principal component analysis (PCA) and sparse singular value decomposition to multi-task learning. In a high-dimensional setting, the classical PCA is inconsistent [73] and recent work has focused on PCA with additional sparse structure on the eigenvectors [5, 14, 18, 24, 125, 92, 142]. Similar sparse structure in singular vectors arises in sparse SVD and biclustering [85, 37, 91, 121, 136, 74, 10, 78, 11]. While the above papers use the sparsity structure of the eigenvectors and singular vectors, it is also possible to have simultaneous low rank and sparse structure directly on the matrix  $\Theta$ . Such a structure arises in multi-task learning, covariance estimation, graph denoising and link prediction [95, 108]. Additional structure on the sparsity pattern was imposed in the context of sparse rank-reduced regression, which is an instance of multi-task learning [38, 20, 93, 9, 111]. Our algorithm described in Section 2 can be applied to the above mentioned problems. In Section 4, we theoretically study multi-task learning in the setting of [93]. We relax conditions imposed in [93], specifically allowing for non-Gaussian errors and not requiring independent samples at each step of the algorithm, while still achieving the near minimax rate of convergence. We provide additional discussion in Section 4 after formally providing results for the multi-task learning setting. In Section 5, we further corroborate our theoretical results in extensive simulations and show that our algorithm outperforms existing methods in multi-task learning.

**Low-rank plus sparse matrix recovery** At this point, it is worth mentioning another commonly encountered structure on the decision variable  $\Theta$  that we do not study in the current paper. In various applications it is common to model  $\Theta$  as a sum of two matrices, one of which is low-rank and the other one sparse. Applications include robust PCA, latent Gaussian graphical models, factor analysis and multi-task learning [27, 67, 33, 41, 2, 56, 143, 133, 57, 60, 31]. While Burer-Monteiro factorization has been considered for the low-rank component

in this context (see, for example, [143] and references therein), the low-rank component is dense as it needs to be incoherent. The incoherence assumption guarantees that the low-rank component is not too spiky and can be identified [29]. An alternative approach was taken in [57] where alternating minimization over the low-rank and sparse component with a projection on a nonconvex set was investigated.

## 1.2. Organization of the paper

In Section 2 we provide details for our proposed algorithm. Section 3 states our assumptions and the theoretical result with a proof sketch. Section 4 shows applications to multi-task learning, while Section 5 presents experimental results. Section 6 provides detailed technical proofs. Conclusion is given in Section 7.

## 2. Gradient descent with hard thresholding

In this section, we detail our proposed algorithm, which is based on gradient descent with hard thresholding (GDT). Our focus is on developing an efficient algorithm for minimizing  $f(\Theta)$  with  $\Theta \in \Xi$ . In statistical estimation and machine learning a common goal is to find  $\Theta^*$ , which is an (approximate) minimizer of  $\mathbb{E}[f(\Theta)]$  where the expectation is with respect to randomness in data. In many settings, the global minimizer of (1.1) can be shown to approximate  $\Theta^*$  up to statistical error, which is problem specific. In Section 3, we will show that iterates of our algorithm converge linearly to  $\Theta^*$  up to a statistical error. It is worth noting that an argument similar to that in the proof of Theorem 1 can be used to establish linear convergence to the global minimizer  $\hat{\Theta}$  in a deterministic setting. That is, suppose  $(\hat{U}, \hat{V})$  is a global minimizer of the problem (1.2) and  $\hat{\Theta} = \hat{U}\hat{V}^\top$ . Then as long as the conditions in Section 3 hold for  $\hat{U}, \hat{V}$  in place of  $U^*, V^*$ , we can show linear convergence to  $\hat{\Theta}$  up to an error level defined by the gradient of the objective function at  $\hat{\Theta}$ . See the discussion after Theorem 1.

Our algorithm, GDT, uses a Burer-Monteiro factorization to write  $\Theta = UV^\top$ , where  $U \in \mathbb{R}^{m_1 \times r}$  and  $V \in \mathbb{R}^{m_2 \times r}$ , and minimizes

$$(\hat{U}, \hat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V) + g(U, V), \quad (2.1)$$

where  $g(U, V)$  is the penalty function defined as

$$g(U, V) = \frac{1}{4} \|U^\top U - V^\top V\|_F^2.$$

The role of the penalty is to find a balanced decomposition of  $\hat{\Theta}$ , one for which  $\sigma_i(\hat{U}) = \sigma_i(\hat{V})$ ,  $i = 1, \dots, r$  [149, 143]. Note the value of the penalty is equal to 0 for a balanced solution, so we can think of the penalized objective as looking through minimizer of (1.2) for a one that satisfies  $\hat{U}^\top \hat{U} - \hat{V}^\top \hat{V} = 0$ . In particular, adding the penalty function  $g$  does not change the minimizer of  $f$  over  $\Xi$ . The

convergence rate of GDT depends on the condition number of  $(U^*, V^*)$ , the point algorithm converges to. The penalty ensures that the iterates  $U, V$  are not ill-conditioned. Gradient descent with hard-thresholding on  $U$  and  $V$  is used to minimize (2.1). Details of GDT are given in Algorithm 1. The algorithm takes as input parameters  $\eta$ , the step size;  $s_1, s_2$ , the sparsity level;  $T$ , the number of iterations; and a starting point  $\Theta^0$ .

The choice of starting point  $\Theta^0$  is very important as the algorithm performs a local search in its neighborhood. In Section 3 we will formalize how close  $\Theta^0$  needs to be to  $\Theta^*$ , while in Section 4 we provide a concrete way to initialize under a multi-task learning model. In general, we envisage finding  $\Theta^0$  by solving the following optimization problem

$$\Theta^0 = \arg \min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) + \text{pen}(\Theta), \quad (2.2)$$

where  $\text{pen}(\Theta)$  is a (simple) convex penalty term making the objective (2.2) a convex optimization problem. For example, we could use the vector  $\ell_1$  norm,  $\text{pen}(\Theta) = \|\Theta\|_1$ . The choice of penalty  $\text{pen}(\Theta)$  should be such that solving the optimization problem in (2.2) can be done efficiently in a high dimensional setting. In practice, if solving the convex relaxation is slow, we can start from the all zero matrix and perform several (proximal) gradient steps to get an appropriate initialization. See for example [143]. Once an initial estimate  $\Theta^0$  is obtained, we find the best rank  $r$  approximation  $\tilde{\Theta} = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$  to  $\Theta^0$  and use it to obtain the initial iterates  $U^0$  and  $V^0$ . In each step, GDT updates  $U$  and  $V$  by taking a gradient step and hard-thresholding the result. The operation  $\text{Hard}(U, s)$  keeps  $s$  rows of  $U$  with the largest  $\ell_2$  row-norm, while setting to zero other rows.

Suppose that the target statistical parameter  $\Theta^*$  is in  $\Xi(r^*, s_1^*, s_2^*)$ . The sparsity level  $s_1^*$  and  $s_2^*$  as well as the rank  $r^*$  are not known in practice, but are needed in Algorithm 1. For the convergence proof we require that the input parameters to the algorithm are set as  $s_1 = c \cdot s_1^*$  and  $s_2 = c \cdot s_2^*$  for some  $c > 1$ . From simulations, we observe that the estimation accuracy is not very sensitive to the choice of  $s_1$  and  $s_2$  as long as they are chosen greater than the true values  $s_1^*$  and  $s_2^*$ . This suggests that in practice, we could set  $s_1$  and  $s_2$  to be reasonably large values whenever a reasonable guess of the sparsity level is available, as incorrectly omitting nonzero value (false negative) is more troublesome than including one zero value (false positive). Alternatively, as we do in simulations, we can use a validation set or an information criteria to select these tuning parameters. However, it is noted in [112] that conventional cross validation may select an inconsistent model, especially when using a non-convex penalty. As an improvement, we can adopt the techniques in [111], which develops the scale-free predictive information criterion to select the best sparsity parameters. Also, [112] proposes structural cross validation method that can achieve the minimax optimal error rate.

Following the same guideline as in the literature, in our analysis we assume that we are using the true rank  $r = r^*$ . In practice, the rank  $r$  can be estimated as in [19], which guarantees consistent rank estimation with high probability.

---

**Algorithm 1** Gradient Descent with Hard Thresholding (GDT)
 

---

1: **Input:** Initial estimate  $\Theta^0$   
 2: **Parameters:** Step size  $\eta$ , Rank  $r$ , Sparsity level  $s_1, s_2$ , Total number of iterations  $T$   
 3:  $(\tilde{U}, \tilde{\Sigma}, \tilde{V}) = \text{rank } r \text{ SVD of } \Theta^0$   
 4:  $U^0 = \text{Hard}(\tilde{U}(\tilde{\Sigma})^{\frac{1}{2}}, s_1)$ ,  $V^0 = \text{Hard}(\tilde{V}(\tilde{\Sigma})^{\frac{1}{2}}, s_2)$   
 5: **for**  $t = 1$  **to**  $T$  **do**  
 6:    $V^{t+0.5} = V^t - \eta \nabla_V f(U^t, V^t) - \eta \nabla_V g(U^t, V^t)$ ,  
 7:    $V^{t+1} = \text{Hard}(V^{t+0.5}, s_2)$   
 8:    $U^{t+0.5} = U^t - \eta \nabla_U f(U^t, V^t) - \eta \nabla_U g(U^t, V^t)$ ,  
 9:    $U^{t+1} = \text{Hard}(U^{t+0.5}, s_1)$   
 10: **end for**  
 11: **Output:**  $\Theta^T = U^T (V^T)^\top$

---

Although [19] considers low-rank structure without sparsity, in practice, it still provides a reasonable rank estimator. The usage of [19] in a low-rank and sparse model is also suggested in [93]. The performance of the GDT algorithm is robust to the choice of rank  $r$ , as we will demonstrate through extensive experiments in Section 5. Finally, we remark that a joint tuning scheme for the rank and sparsity parameters can also be considered.

To the best of our knowledge, GDT is the first gradient based algorithm to deal with a nonconvex optimization problem over a parameter space that is simultaneously low rank and row and column sparse. In the following section we will provide conditions on the objective function  $f$  and the starting point  $\Theta^0$  which guarantee linear convergence to  $\Theta^*$  up to a statistical error. As an application, we consider the multi-task learning problem in Section 4. We show that the statistical error nearly matches the optimal minimax rate, while the algorithm achieves the best performance in terms of estimation and prediction error in simulations.

### 3. Theoretical result

In this section, we formalize the conditions and state the main result on the linear convergence of our algorithm. We begin in Section 3.1 by stating the conditions on the objective function  $f$  and initialization that are needed for our analysis. In Section 3.2, we state Theorem 1 that guarantees linear convergence under the conditions to a statistically useful point. The proof outline is given in Section 3.3. In Section 4 to follow, we derive results for multi-task learning as corollaries of our main result.

#### 3.1. Regularity conditions

We start by stating mild conditions on the objective function  $f$ , which have been used in the literature on high-dimensional estimation and nonconvex optimization, and they hold with high-probability for a number of statistical models of interest [144, 143, 57]. Note that all the conditions depend on the choice of  $s_1$  and  $s_2$  (or equivalently, on  $c$ ).



For  $\Theta^* \in \Xi(r^*, s_1^*, s_2^*)$ , let  $\Theta^* = U_{\Theta^*} \Sigma_{\Theta^*} V_{\Theta^*}^\top$  be its singular value decomposition. Let  $U^* = U_{\Theta^*} \Sigma_{\Theta^*}^{1/2}$  and  $V^* = V_{\Theta^*} \Sigma_{\Theta^*}^{1/2}$  be the balanced decomposition of  $\Theta^* = U^* V^{*\top}$ . Note that the decomposition is not unique as  $\Theta^* = (U^* O)(V^* O)^\top$  for any orthogonal matrix  $O \in \mathcal{O}(r)$ . Let  $\sigma_1(\Theta^*) = \sigma_{\max}(\Theta^*)$  and  $\sigma_r(\Theta^*) = \sigma_{\min}(\Theta^*)$  denote the maximum and minimum nonzero singular values of  $\Theta^*$  with  $r = r^*$ . The first condition is Restricted Strong Convexity and Smoothness on  $f$ .

**Restricted Strong Convexity and Smoothness (RSC/RSS)** There exist universal constants  $\mu$  and  $L$  such that

$$\frac{\mu}{2} \|\Theta_2 - \Theta_1\|_F^2 \leq f(\Theta_2) - f(\Theta_1) - \langle \nabla f(\Theta_1), \Theta_2 - \Theta_1 \rangle \leq \frac{L}{2} \|\Theta_2 - \Theta_1\|_F^2$$

for all  $\Theta_1, \Theta_2 \in \Xi(2r, \tilde{s}_1, \tilde{s}_2)$  where  $\tilde{s}_1 = (2c+1)s_1^*$  and  $\tilde{s}_2 = (2c+1)s_2^*$ .

The next condition is on the initial estimate  $\Theta^0$ . It quantifies how close the initial estimator needs to be to  $\Theta^*$  so that iterates of GDT converge to statistically useful solution.

**Initialization (I)** Define  $\mu_{\min} = \frac{1}{8} \min\{1, \frac{\mu L}{\mu+L}\}$  and

$$I_0 = \frac{4}{5} \mu_{\min} \sigma_r(\Theta^*) \cdot \min\left\{\frac{1}{\mu+L}, 2\right\}. \quad (3.1)$$

We require

$$\|\Theta^0 - \Theta^*\|_F \leq \frac{1}{5} \min\left\{\sigma_r(\Theta^*), \frac{I_0}{\xi} \sqrt{\sigma_r(\Theta^*)}\right\}, \quad (3.2)$$

where  $\xi^2 = 1 + \frac{2}{\sqrt{c-1}}$ .

We note that, in general, (3.2) defines a ball of constant radius around  $\Theta^*$  in which the initial estimator needs to fall into. In particular, when considering statistical learning problems, the initial estimator can be inconsistent as the sample size increases.

Next, we define the notion of the statistical error,

$$e_{\text{stat}} = \sup_{\substack{\Delta \in \Xi(2r, \tilde{s}_1, \tilde{s}_2) \\ \|\Delta\|_F \leq 1}} \langle \nabla f(\Theta^*), \Delta \rangle. \quad (3.3)$$

Note that the statistical error quantifies how large the gradient of the objective evaluated at the true parameter  $\Theta^*$  can be in the directions of simultaneously low-rank and sparse matrices. It implicitly depends on the choice of  $c$  and as we will see later there is a trade-off in balancing the statistical error and convergence rate of GDT. As  $c$  increases, statistical error gets larger, but requires us to choose a smaller step size in order to guarantee convergence.

With these two conditions, we are ready to the choice of the step size in Algorithm 1.

**Step size selection** Let  $Z^0 = \begin{bmatrix} U^0 \\ V^0 \end{bmatrix}$ . We choose the step size  $\eta$  to satisfy

$$\eta \leq \frac{1}{16\|Z^0\|_2^2} \cdot \min \left\{ \frac{1}{2(\mu + L)}, 1 \right\}, \quad (3.4)$$

Furthermore, we require  $\eta$  and  $c$  to satisfy

$$\beta = \xi^2 \left( 1 - \eta \cdot \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \right) < 1, \quad (3.5)$$

and

$$e_{\text{stat}}^2 \leq \frac{1 - \beta}{\xi^2 \eta} \cdot \frac{L\mu}{L + \mu} \cdot I_0^2. \quad (3.6)$$

The condition that the step size  $\eta$  satisfies (3.4) is typical in the literature on convex optimization of strongly convex and smooth functions. Under (3.5) we will be able to show contraction after one iteration and progress towards  $\Theta^*$ . The second term in (3.5) is always smaller than 1, while the first term  $\xi^2$  is slightly larger than 1 and is the price we pay for the hard thresholding step. In order to show linear convergence we need to balance the choice of  $\eta$  and  $\xi^2$  to ensure that  $\beta < 1$ . From (3.5), we see that if we select a small step size  $\eta$ , then we need to have a small  $\xi^2$ , which means a large  $c$ . Intuitively, if  $\eta$  is too small, it may be impossible to change row and column support in each iteration. In this case we have to keep many active rows and columns to make sure we do not miss the true signal. This leads to large  $s_1$  and  $s_2$ , or equivalently to a large  $c$ . However, the statistical error (3.3) will increase with increase of  $c$  and these are the trade-off on the selection of  $\eta$  and  $c$ .

Finally, (3.6) guarantees that the iterates do not run outside of the initial ball given in (3.2). In case (3.6) is violated, then the initialization point  $\Theta^0$  is already a good enough estimate of  $\Theta^*$ . Therefore, this requirement is not restrictive. In practice, we found that the selection of  $\eta$  and  $c$  is not restrictive and the convergence is guaranteed for a wide range of values of their values.

In order to satisfy these regularity conditions, we may need to choose a relatively large  $c$ . However, the condition on  $c$  is purely a technical conditions. To the best of our knowledge, all the literature on iterative hard thresholding requires some restrictive conditions on  $c$ . Without the hard thresholding step, we can guarantee contraction  $\beta < 1$  after one step of the gradient descent. However, the hard thresholding step amplifies the estimation error and, therefore, we need a relatively large  $c$  to guarantee contraction. In theory, we require an upper bound on  $c$  that does not scale with  $n, p$ , or  $K$ . In practice, we do not know the true sparsity level  $s^*$  and choose  $s$  directly based on prior knowledge or select it via cross validation. Moreover, the step size  $\eta$  could be selected in a heuristic way when implementing the algorithm for specific applications. While the techniques needed to establish better theoretical control of the parameter  $c$  still require improvement, in practice, even with small values of  $c$  the method performs well. Experiments in Section 5 show that the tuning parameters can be chosen in way to yield good finite sample performance. In practice, selecting

inappropriate model parameters or initialization may worsen the performance of the algorithm, resulting in possibly sublinear convergence rate.

### 3.2. Main result

Our main result establishes linear convergence of GDT iterates to  $\Theta^*$  up to statistical error. Since the factorization of  $\Theta^*$  is not unique, we turn to measure the subspace distance of the iterates  $(U^t, V^t)$  to the balanced decomposition  $U^*(V^*)^\top = \Theta^*$ .

**Subspace distance** Let  $Z^* = \begin{bmatrix} U^* \\ V^* \end{bmatrix}$  where  $\Theta^* = U^*V^{*\top}$  and  $\sigma_i(U^*) = \sigma_i(V^*)$  for each  $i = 1, \dots, r$ . Define the subspace distance between  $Z = \begin{bmatrix} U \\ V \end{bmatrix}$  and  $Z^* = \begin{bmatrix} U^* \\ V^* \end{bmatrix}$  as

$$d^2(Z, Z^*) = \min_{O \in \mathcal{O}(r)} \{ \|U - U^*O\|_F^2 + \|V - V^*O\|_F^2 \}.$$

With this, we are ready to state our main result.

**Theorem 1.** *Suppose the conditions (RSC/RSS), (I) are satisfied and the step size  $\eta$  satisfies (3.4)–(3.6). Then after  $T$  iterations of GDT (Algorithm 1), we have*

$$d^2(Z^T, Z^*) \leq \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2. \quad (3.7)$$

Furthermore, for  $\Theta^T = U^T(V^T)^\top$  we have

$$\|\Theta^T - \Theta^*\|_F^2 \leq 4\sigma_1(\Theta^*) \cdot \left[ \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2 \right]. \quad (3.8)$$

The proof sketch of Theorem 1 is given in the following section. Conceptually, Theorem 1 provides a minimal set of conditions for convergence of GDT. The first term in equations (3.7) and (3.8) correspond to the optimization error, whereas the second term corresponds to the statistical error. These bounds show that the distance between the iterates and  $\Theta^*$  drop exponentially up to the statistical limit  $e_{\text{stat}}$ , which is problem specific. In statistical learning problem, it commonly depends on the sample size and the signal-to-noise ratio of the problem.

Theorem 1 provides convergence in a statistical setting to the “true” parameter  $\Theta^*$ . However, as mentioned in Section 2, Algorithm 1 and Theorem 1 can also be used to establish linear convergence to a global minimizer in a deterministic setting. Suppose  $(\widehat{U}, \widehat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} \{f(U, V)\}$  is a global minimizer and  $\widehat{\Theta} = \widehat{U}\widehat{V}^\top$ . Furthermore, assume that the conditions in Section 3.1 are satisfied with  $\widehat{\Theta}$  in place of  $\Theta^*$ . Then we have that the iterates  $\{\Theta^t\}$  obtained by GDT converge linearly to a global minimum  $\widehat{\Theta}$  up to the error  $\widehat{e}_{\text{stat}}$  defined similar to

(3.3) with  $\widehat{\Theta}$  in place of  $\Theta^*$ . This error comes from sparsity and hard thresholding. In particular, suppose there are no row or column sparsity constraints in the optimization problem (1.2), so that we do not have hard-thresholding steps in Algorithm 1. Then we have  $\widehat{e}_{\text{stat}} = 0$ , so that iterates  $\{\Theta^t\}$  converge linearly to  $\widehat{\Theta}$ , recovering the result of [144].

### 3.3. Proof sketch of Theorem 1

In this section we sketch the proof of our main result. The proof combines three lemmas. We first one quantify the accuracy of the initialization step. The following one quantifies the improvement in the accuracy by one step of GDT. The third lemma shows that the step size assumed in Theorem 1 satisfies conditions of the second lemma. Detailed proofs of these lemmas are relegated to Section 6.

Our first lemma quantifies the accuracy of the initialization step.

**Lemma 2.** *Suppose that the input to GDT,  $\Theta^0$ , satisfies initialization condition (3.2). Then the initial iterates  $U^0$  and  $V^0$  obtained in lines 3 and 4 of Algorithm 1 satisfy*

$$d(Z^0, Z^*) \leq I_0, \quad (3.9)$$

where  $Z^0 = \begin{bmatrix} U^0 \\ V^0 \end{bmatrix}$  and  $I_0$  is defined in (3.1).

The proof of Lemma 2 is given in Section 6.1.

**Lemma 3.** *Suppose the conditions (RSC/RSS), (I) are satisfied. Assume that the point  $Z = \begin{bmatrix} U \\ V \end{bmatrix}$  satisfies  $d(Z, Z^*) \leq I_0$ . Let  $(U^+, V^+)$  denote the next iterate obtained with Algorithm 1 with the step size  $\eta$  satisfying*

$$\eta \leq \frac{1}{8\|Z\|_2^2} \cdot \min \left\{ \frac{1}{2(\mu + L)}, 1 \right\}. \quad (3.10)$$

Then we have

$$d^2(Z^+, Z^*) \leq \xi^2 \left[ \left( 1 - \eta \cdot \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \right) \cdot d^2(Z, Z^*) + \eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2 \right], \quad (3.11)$$

where  $\xi^2 = 1 + \frac{2}{\sqrt{c-1}}$ .

The proof of Lemma 3 is given in Section 6.2.

**Lemma 4.** *Suppose  $Z = \begin{bmatrix} U \\ V \end{bmatrix}$  satisfies  $d(Z, Z^*) \leq I_0$ . We have that the choice of step size (3.4) in Theorem 1 satisfies the condition (3.10) in Lemma 3.*

The proof of Lemma 4 is given in Section 6.3.

Combining the three results above, we can complete the proof of Theorem 1. Starting from initialization  $\Theta^0$  satisfying the initialization condition (3.2),

Lemma 2 ensures that (3.9) is satisfied for  $Z^0$  and Lemma 4 ensures that the choice of step size (3.4) satisfies the step size condition (3.10) in Lemma 3. We can then apply Lemma 3 and get the next iterate  $Z^1 = Z^+$ , which satisfies (3.11). Using the condition on statistical error (3.6), initialization (3.2), and a simple calculation, we can verify that  $Z^1$  satisfies  $d(Z^1, Z^*) \leq I_0$ . Therefore we can apply Lemma 2, Lemma 3, and Lemma 4 repeatedly to obtain

$$d^2(Z^{t+1}, Z^*) \leq \beta \cdot d^2(Z^t, Z^*) + \xi^2 \eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2,$$

for each  $t = 0, 1, \dots, T$ . We then have

$$d^2(Z^T, Z^*) \leq \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2.$$

Finally, for  $\Theta^T = U^T(V^T)^\top$ , let  $O^T \in \mathcal{O}(r)$  be such that

$$d^2(Z^T, Z^*) = \|U^T - U^*O^T\|_F^2 + \|V^T - V^*O^T\|_F^2.$$

We have

$$\begin{aligned} \|\Theta^T - \Theta^*\|_F^2 &= \|U^T(V^T)^\top - U^*O^T(V^*O^T)^\top\|_F^2 \\ &\leq \left[ \|U^T\|_2 \|V^T - V^*O^T\|_F + \|V^*\|_2 \|U^T - U^*O^T\|_F \right]^2 \\ &\leq \|U^T\|_2^2 \|V^T - V^*O^T\|_F^2 + \|V^*\|_2^2 \|U^T - U^*O^T\|_F^2 \\ &\leq 2\|Z^*\|_2^2 \cdot d^2(Z^T, Z^*) \\ &\leq 4\sigma_1(\Theta^*) \cdot \left[ \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2 \right], \end{aligned}$$

which shows linear convergence up to the statistical error.

#### 4. Application to multi-task learning

In this section, we apply the theory developed in Section 3 on two specific problems. First, in Section 4.1, we apply GDT algorithm to a multi-task learning problem. We show that under commonly used statistical conditions the conditions on the objective function stated in Section 3.1 are satisfied with high-probability. Next, in Section 4.2 we discuss an application to multi-task reinforcement learning problem.

##### 4.1. GDT for multi-task learning

We apply GDT algorithm to the problem of multi-task learning, which has been successfully applied in a wide range of application areas, ranging from neuroscience [123], natural language understanding [44], speech recognition [110],

computer vision [111], and genetics [139] to remote sensing [134], image classification [82], spam filtering [130], web search [34], disease prediction [146], and eQTL mapping [77]. By transferring information between related tasks it is hoped that samples will be better utilized, leading to improved generalization performance.

We consider the following linear multi-task learning problem

$$Y = X\Theta^* + E, \quad (4.1)$$

where  $Y \in \mathbb{R}^{n \times k}$  is the response matrix,  $X \in \mathbb{R}^{n \times p}$  is the matrix of predictors,  $\Theta^* \in \mathbb{R}^{p \times k}$  is an unknown matrix of coefficients, and  $E \in \mathbb{R}^{n \times k}$  is an unobserved noise matrix with i.i.d. mean zero and variance  $\sigma^2$  entries. Here  $n$  denotes the sample size,  $k$  is the number of responses, and  $p$  is the number of predictors. In general multi-task learning problems, the design matrix  $X$  may be different for different tasks. Throughout the paper we assume a common design matrix  $X$  for simplicity, and it is straightforward to generalize the result to problem with different  $X$  for different tasks.

There are a number of ways to capture relationships between different tasks and success of different methods relies on this relationship. [48] studied a setting where linear predictors are close to each other. In a high-dimensional setting, with large number of variables, it is common to assume that there are a few variables predictive of all tasks, while others are not predictive [120, 102, 90, 79, 127]. Another popular condition is to assume that the predictors lie in a shared lower dimensional subspace [7, 6, 140, 8, 126]. In contemporary applications, however, it is increasingly common that both the number of predictors and the number of tasks is large compared to the sample size. For example, in a study of regulatory relationships between genome-wide measurements, where micro-RNA measurements are used to explain the gene expression levels, it is commonly assumed that a small number of micro-RNAs regulate genes participating in few regulatory pathways [91]. In such a setting, it is reasonable to assume that the coefficients are both sparse and low rank. That is, one believes that the predictors can be combined into fewer latent features that drive the variation in the multiple response variables and are composed only of relevant predictor variables. Compared to a setting where either variables are selected or latent features are learned, there is much less work on simultaneous variable selection and rank reduction [19, 37, 38, 111]. In addition, when both  $p$  and  $k$  are large, it is also needed to assume the column sparsity on the matrix  $\Theta^*$  to make estimation feasible [93], a model that has been referred to as the two-way sparse reduced-rank regression model. We focus on this model here.

**Multi-task Model (MTM)** In the model (4.1), we assume that the true coefficient matrix  $\Theta^* \in \Xi(r, s_1^*, s_2^*)$ . The noise matrix  $E$  has i.i.d. sub-Gaussian elements with variance proxy  $\sigma^2$ , which requires that each element  $e_{ij}$  satisfies  $\mathbb{E}(e_{ij}) = 0$  and its moment generating function satisfies  $\mathbb{E}[\exp(te_{ij})] \leq \exp(\sigma^2 t^2/2)$ . The design matrix  $X$  is considered fixed with columns normalized to have mean 0 and standard deviation 1. Moreover, we assume  $X$  satisfies the

following Restricted Eigenvalue (RE) condition [100] for some constant  $\underline{\kappa}(s_1)$  and  $\bar{\kappa}(s_1)$ .

$$\underline{\kappa}(s_1) \cdot \|\theta\|_2^2 \leq \frac{1}{n} \|X\theta\|_2^2 \leq \bar{\kappa}(s_1) \cdot \|\theta\|_2^2 \quad \text{for all } \|\theta\|_0 \leq s_1.$$

We will show that under the condition **(MTM)**, GDT converges linearly to the optimal coefficient  $\Theta^*$  up to a region of statistical error. Compared to the previous methods for estimating jointly sparse and low rank coefficients [19, 37, 38, 93], GDT is more scalable and improves estimation accuracy as illustrated in the simulation Section 5.

In the context of the multi-task learning with the model in (4.1), we are going to use the least squares loss. The objective function in is  $f(\Theta) = \frac{1}{2n} \|Y - X\Theta\|_F^2$  and we write  $\Theta = UV^\top$  with  $U \in \mathbb{R}^{p \times r}$  and  $V \in \mathbb{R}^{k \times r}$ . The constraint set is set as before as  $U \in \mathcal{U}(s_1)$  and  $V \in \mathcal{U}(s_2)$  with  $s_1 = c \cdot s_1^*$ ,  $s_2 = c \cdot s_2^*$  for some  $c > 1$ . The rank  $r$  and the sparsity levels  $s_1, s_2$  are tuning parameters, which can be selected using the information criterion as in [111].

In order to apply the results of Theorem 1, we first verify the conditions in Section 3.1. The condition **(RSC/RSS)** in is equivalent to

$$\mu \|\Theta_2 - \Theta_1\|_F^2 \leq \left\langle \frac{1}{n} X^\top X (\Theta_2 - \Theta_1), \Theta_2 - \Theta_1 \right\rangle \leq L \|\Theta_2 - \Theta_1\|_F^2,$$

and it holds with  $\mu = \underline{\kappa}(s_1)$  and  $L = \bar{\kappa}(s_1)$ .

Next, we discuss how to initialize GDT in the context of multi-task learning. Under the structural conditions on  $\Theta^*$  in the condition **(MTM)** there are a number of way to obtain an initial estimator  $\Theta^0$ . For example, we can use row and column screening [50], group lasso [141], and lasso [118] among other procedures. Here and in simulations we use the lasso estimator, which takes the form

$$\Theta^0 = \arg \min_{\Theta \in \mathbb{R}^{p \times k}} \frac{1}{2n} \|Y - X\Theta\|_F^2 + \lambda \|\Theta\|_1.$$

The benefit of this approach is that it is scalable to the high-dimensional setting and trivially parallelizable, since each column of  $\Theta^0$  can be estimated separately. The requirement of the initialization condition **(I)** is effectively a requirement on the sample size. Under the condition **(MTM)**, a result of [100] shows that these conditions are satisfied with  $n \geq s_1^* s_2^* \log p \log k$ .

We then characterize the statistical error  $e_{\text{stat}}$  under the condition **(MTM)**.

**Lemma 5.** *Under the condition **(MTM)**, with probability at least  $1 - (p \vee k)^{-1}$  we have*

$$e_{\text{stat}} \leq C\sigma \sqrt{\frac{(s_1^* + s_2^*)(r + \log(p \vee k))}{n}}$$

for some constant  $C$ .

The proof of Lemma 5 is given in Section 6.4.

With these conditions, we have the following result on GDT when applied to the multi-task learning model in (4.1).

**Corollary 6.** *Suppose that the condition (MTM) is satisfied and the step size  $\eta$  satisfies (3.4)–(3.6). Then for all*

$$T \geq C \log \left[ \frac{n}{(s_1^* + s_2^*)(r + \log(p \vee k))} \right],$$

with probability at least  $1 - (p \vee k)^{-1}$ , we have

$$\|\Theta^T - \Theta^*\|_F \leq C\sigma \sqrt{\frac{(s_1^* + s_2^*)(r + \log(p \vee k))}{n}}$$

for some constant  $C$ .

Each iteration of the algorithm requires computing the gradient step with time complexity  $r(n+r)(p+k)$ . Note that if there is no error term  $E$  in the model (4.1), then Algorithm 1 converges linearly to the true coefficient matrix  $\Theta^*$ , since  $e_{\text{stat}} = 0$  in that case. The error rate in Corollary 6 matches the error rate of the algorithm proposed in [93]. However, our algorithm does not require a new independent sample in each iteration and allows for non-Gaussian errors. Compared to the minimax rate

$$\sigma \sqrt{\frac{1}{n} \left[ (s_1^* + s_2^*)r + s_1^* \log \frac{ep}{s_1^*} + s_2^* \log \frac{ek}{s_2^*} \right]} \quad (4.2)$$

established in [93], both our algorithm and that of [93] match the rate up to a multiplicative log factor. To the best of our knowledge, achieving the minimax rate (4.2) with a computationally scalable procedure is still an open problem. Note, however, that when  $r$  is comparable to  $\log(p \vee k)$  the rates match up to a constant multiplier. Therefore for large enough  $T$ , GDT algorithm attains near optimal rate.

In case we do not consider column sparsity, that is, when  $s_2^* = k$ , Corollary 6 gives error rate

$$\|\Theta^T - \Theta^*\|_F \leq C\sigma \sqrt{\frac{kr + s_1^*(r + \log p)}{n}} \quad (4.3)$$

and prediction error

$$\|X\Theta^T - X\Theta^*\|_F^2 \leq C\sigma^2 (kr + s_1^*(r + \log p)).$$

Compared to the prediction error bound  $kr + s_1^*r \log \frac{p}{s}$  proved in [20], we see that GDT error is much smaller with  $r + \log p$  in place of  $r \log p$ . Moreover, GDT error matches the prediction error  $(k + s_1^* - r)r + s_1^* \log p$  established in [111], as long as  $k \geq Cr$  which is typically satisfied.

As mentioned before, in practice we use the criterion in [19] to select the rank  $r$ . In order to obtain a consistent rank estimator with high probability, the procedure in [19] requires that  $\sigma_{r^*}(X\Theta^*)$  is lower bounded. We emphasize that although this condition is not required to obtain a near minimax optimal statistical error, it does affect the convergence rate of the GDT algorithm.



#### 4.2. Application to multi-task reinforcement learning

Reinforcement learning (RL) and approximate dynamic programming (ADP) are popular algorithms that help decision makers find optimal policies for decision making problems under uncertainty that can be cast in the framework of Markov Decision Processes (MDP) [15, 116]. Similar to many other approaches, when the sample size is small these algorithms may have poor performance. A possible workaround then is to simultaneously solve multiple related tasks and take advantage of their similarity and shared structure. This approach is called multi-task reinforcement learning (MTRL) and has been studied extensively [83, 131, 113]. In this section we show how GDT algorithm can be applied to the MTRL problem.

A Markov decision process (MDP) is represented by a 5-tuple  $\mathcal{M} = (S, A, P, R, \gamma)$  where  $S$  represents the state space (which we assume to be finite for simplicity);  $A$  is a finite set of actions;  $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$  is the Markovian transition kernel that measures the probability that action  $a$  in state  $s$  at time  $t$  will lead to state  $s'$  at time  $t + 1$  (we assume  $P_a$  to be time homogeneous);  $R(s, a)$  is the state-action reward function measuring the instantaneous reward received when taking action  $a$  in state  $s$ ; and  $\gamma$  is the discount factor. The core problem of MDP is to find a deterministic policy  $\pi : S \rightarrow A$  that specifies the action to take when decision maker is in some state  $s$ . Define the Bellman operator

$$\mathcal{T}Q(s, a) = R(s, a) + \gamma \sum_{s'} P_a(s, s') \max_{a'} Q(s', a'),$$

where  $Q : S \times A \rightarrow \mathbb{R}$  is the state-action value function. The MDP can then be solved by calculating the optimal state-action value function  $Q^*$  which gives the total discounted reward obtained starting in state  $s$  and taking action  $a$ , and then following the optimal policy in subsequent time steps. Given  $Q^*$ , the optimal policy is recovered by the greedy policy:  $\pi^*(s) = \arg \max_{a \in A} Q^*(s, a)$ .

In MTRL the objective is to solve  $k$  related tasks simultaneously where each task  $k_0 \in \{1, \dots, k\}$  corresponds to an MDP:  $\mathcal{M}_{k_0} = (S, A, P_{k_0}, R_{k_0}, \gamma_{k_0})$ . Thus, these  $k$  tasks share the same state and action space but each task has a different transition dynamics  $P_{k_0}$ , state-action reward function  $R_{k_0}$ , and discount factor  $\gamma_{k_0}$ . The decision maker's goal is to find an optimal policy for each MDP. If these MDPs do not share any information or structure, then it is straightforward to solve each of them separately. Here we assume the MDPs do share some structure so that the  $k$  tasks can be learned together with smaller sample complexity than learning them separately.

We follow the structure in [26] and solve this MTRL problem by the fitted- $Q$  iteration (FQI) algorithm [47], one of the most popular method for ADP. In contrast to exact value iteration ( $Q^t = \mathcal{T}Q^{t-1}$ ), in FQI this iteration is approximated by solving a regression problem by representing  $Q(s, a)$  as a linear function in some features representing the state-action pairs. To be more specific, we denote  $\varphi(s) = [\varphi_1(s), \varphi_2(s), \dots, \varphi_{p_s}(s)]$  as the feature mapping for state  $s$

**Algorithm 2** Multi-Task Reinforcement Learning with GDT

---

**Input:** States  $S_k = \{s_i\}_{i=1}^{n_s} \subseteq S$ .  
**Initialize**  $\Theta^0 = 0$   
**for**  $t = 1$  **to**  $T$  **do**  
  **for**  $a = 1$  **to**  $|A|$  **do**  
    **for**  $k_0 = 1$  **to**  $k$ ,  $i = 1$  **to**  $n_s$  **do**  
      Generate samples  $r_{i,a,k_0}^t = R_{k_0}(s_i, a)$  and  $s'_{i,a,k_0}^t \sim P_{a,k_0}(s_i, s')$   
      Calculate  $y_{i,a,k_0}^t = r_{i,a,k_0}^t + \gamma \max_{a'} \widehat{Q}_{k_0}^{t-1}(s'_{i,a,k_0}^t, a')$   
    **end for**  
  **end for**  
  Estimate  $\Theta^t$  using GDT algorithm with  $X = \{X((s_i, a), \cdot) = \phi(s_i, a)^\top\}_{s_i \in S_k, a \in A}$  and  
 $Y = \{Y((s_i, a), k_0) = y_{i,a,k_0}^t\}_{s_i \in S, a \in A, k_0 \in [k]}$   
**end for**  
**Output:**  $\Theta^T$

---

where  $\varphi_i : S \rightarrow \mathbb{R}$  denotes the  $i$ th feature. We then extend the state-feature vector  $\varphi$  to a feature vector mapping state  $s$  and action  $a$  as:

$$\phi(s, a) = [ \underbrace{0, 0, \dots, 0}_{(a-1) \times p_s \text{ times}}, \varphi_1(s), \varphi_2(s), \dots, \varphi_{p_s}(s), \underbrace{0, 0, \dots, 0}_{(|A|-a) \times p_s \text{ times}} ] \in \mathbb{R}^p,$$

where  $p = |A| \times p_s$ . Finally, for MDP  $k_0$ , we represent the state-action value function  $Q_{k_0}(\cdot, \cdot)$  as an  $|S| \times |A|$  dimensional column vector with:

$$Q_{k_0}(s, a) = \phi(s, a)^\top \cdot \Theta_{k_0}$$

where  $\Theta_{k_0}$  is a  $p \times 1$  dimensional column vector. If  $\Theta \in \mathbb{R}^{p \times k}$  represents the matrix with columns  $\Theta_{k_0}$ ,  $k \in \{1, \dots, k\}$ , then we see that given the  $Q_{k_0}(s, a)$  state-action value functions, estimating the  $\Theta$  matrix is just a Multi-Task Learning problem of the form (4.1) with the response matrix  $Y \doteq Q \in \mathbb{R}^{n \times k}$  where  $n = |S| \times |A|$  denotes the ‘‘sample size’’ with rows indexed by pairs  $(s, a) \in S \times A$ ,  $X \doteq \Phi \in \mathbb{R}^{n \times p}$  represents the matrix of predictors (features) with  $(s, a)^{th}$  row as  $\phi(s, a)$ , and  $\Theta^*$  is the unknown matrix of ADP coefficients. Consistent with the GDT algorithm, to exploit shared sparsity and structure across the  $k$  MDP tasks, we will subsequently assume that the coefficient matrix  $\Theta^*$  is row sparse and low rank.

Algorithm 2 provides details of MTRL with GDT. We assume we have access to the generative model of the  $k$  MDPs so that we can sample reward  $r$  and state  $s'$  from  $R(s, a)$  and  $P_a(s, s')$ . With ‘‘design states’’  $S_k \subseteq S$ ,  $n_s \doteq |S_k|$  given as input, for each action  $a$  and each state  $s \in S_k$ , FQI first generates samples (reward  $r$  and transition state  $s'$ ) from the generative model of each MDP. These samples form a new dataset according to

$$y_{i,a,k_0}^t = r_{i,a,k_0}^t + \gamma \max_{a'} \widehat{Q}_{k_0}^{t-1}(s'_{i,a,k_0}^t, a').$$

Here  $\widehat{Q}_{k_0}^{t-1}$  is calculated using the coefficient matrix from previous iteration:

$$\widehat{Q}_{k_0}^{t-1}(s'_{i,a,k_0}^t, a') = \phi(s'_{i,a,k_0}^t, a')^\top \cdot \Theta_{k_0}^{t-1}$$

We then build dataset  $\mathcal{D}_{k_0}^t = \{(s_i, a), y_{i,a,k_0}^t\}_{s_i \in S_k, a \in A}$  with  $s$  as predictor and  $y$  as response, and apply GDT algorithm on the dataset  $\{D_{k_0}^t\}_{k_0=1}^k$  to get estimator  $\Theta^t$ . This completes an iteration  $t$  and we repeat this process until convergence. Finally the optimal policy  $\pi_{k_0}^t$  is given by greedy policy:  $\pi_{k_0}^t(s) = \arg \max_{a \in A} \widehat{Q}_{k_0}^t(s, a)$  at each iteration  $t$ .

To derive theoretical result analogous to [26], we further assume  $R(s, a) \in [0, 1]$  and hence the maximum cumulative discounted reward  $Q_{\max} = 1/(1-\gamma)$ . Since each task is a meaningful MDP, we do not assume sparsity on columns. Suppose  $\sup_s \|\varphi(s)\|_2 \leq L$ , we have the following theoretical result:

**Theorem 7.** *Suppose the linear model holds and suppose the conditions in Section 3 are satisfied for each  $\Theta_a^*$  with rank  $r$  and row sparsity  $s_1^*$ , then after  $T$  iterations, with probability at least  $(1 - (p \wedge k)^{-1})^T$  we have*

$$\begin{aligned} \frac{1}{k} \sum_{k_0=1}^k \left\| Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T} \right\|_2^2 &\leq \frac{C}{(1-\gamma)^4} \left[ \frac{1}{n} Q_{\max}^2 L^4 \left( r + \frac{s_1^*}{k} (r + \log p) \right) \right] \\ &\quad + \frac{4Q_{\max}^2}{(1-\gamma)^4} \left[ C\beta^T + \gamma^T \right]^2 \end{aligned}$$

for some constant  $C$ .

*Proof.* We start from the intermediate result in [96]:

$$\left| Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T} \right| \leq \frac{2\gamma(1-\gamma^{T+1})}{(1-\gamma)^2} \left[ \sum_{t=0}^{T-1} \alpha_t |\epsilon_{k_0}^t| + \alpha_T |Q_t^* - Q_t^0| \right],$$

where

$$\alpha_t = \frac{(1-\gamma)\gamma^{T-t-1}}{1-\gamma^{T+1}}, \text{ for } t < T, \text{ and } \alpha_T = \frac{(1-\gamma)\gamma^T}{1-\gamma^{T+1}}.$$

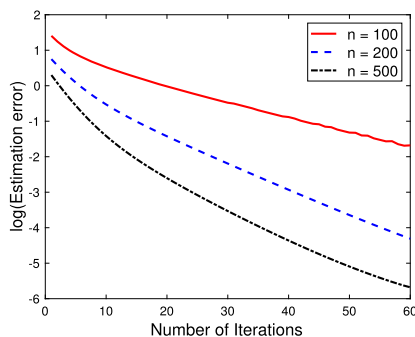
The error term  $\epsilon_{k_0}^t(s', b)$  measures the approximation error in state  $s' \in S$  and action  $b \in A$ . It can be bounded by

$$\begin{aligned} |\epsilon_{k_0}^t(s', b)| &= |\varphi(s')^\top \Theta_{k_0,b}^t - \varphi(s')^\top \Theta_{k_0,b}^*| \leq \|\varphi(s')\|_2 \|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\|_2 \\ &\leq L \|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\|_2. \end{aligned}$$

We then have

$$\left| Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T} \right| \leq \frac{2\gamma(1-\gamma^{T+1})}{(1-\gamma)^2} \left[ \sum_{t=0}^{T-1} \alpha_t L \max_b \|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\|_2 + 2\alpha_T Q_{\max} \right].$$

Taking average, and plugging in the main result (3.8) and the statistical error (4.3) we obtain our desired result.  $\square$

FIG 1. *No error case*

## 5. Experiment

In this section we demonstrate the effectiveness of the GDT algorithm by extensive experiments.<sup>1</sup> Section 5.1 shows results on synthetic datasets while Section 5.2 and 5.3 show results on two real datasets.

### 5.1. Synthetic datasets

We present numerical experiments on MTL problem to support our theoretical analysis. Throughout this section, we generate the instances by sampling all entries of design matrix  $X$ , all nonzero entries of the true signal  $U^*$  and  $V^*$ , and all entries of the noise matrix  $E$  as i.i.d. standard normal.

**Linear convergence** We first demonstrate our linear convergence result. Because it is hard to quantify linear convergence with statistical error, we turn to show the linear convergence in some special cases. Firstly, as we discussed after Corollary 6, suppose there is no error term  $E$  in the model (4.1), then Algorithm 1 converges linearly to the true coefficient matrix  $\Theta^*$ . In this case we choose  $p = 100, k = 50, r = 8, s_1^* = s_2^* = 10$ , and the estimation error is shown in Figure 1. Secondly, as we discussed at the end of Section 3.2, suppose there are no row or column sparsity constraints on  $\Theta^*$ , then Algorithm 1 converges linearly to global minimum  $\hat{\Theta}$ . In this case it is more likely that we are in low dimensions, therefore we choose  $p = 50$ . The estimation error is shown in Figure 2. We see that in both cases GDT has linear convergence rate.

**Estimation accuracy** We compare our algorithm with the Double Projected Penalization (DPP) method in [93], the thresholding SVD method (TSVD) method in [91], the exclusive extraction algorithm (EEA) in [37], the two methods (denoted by RCGL and JRRS) in [20], and the standard Multitask learning method (MTL, with  $L_{2,1}$  penalty). Here we set  $n = 50, p = 100, k = 50, r =$

<sup>1</sup>The codes are available at [https://github.com/ming93/GDT\\_nonconvex](https://github.com/ming93/GDT_nonconvex).

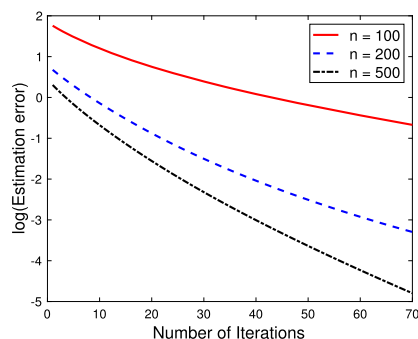


FIG 2. No sparsity case

8,  $s_1^* = s_2^* = 10$ . The reason why we choose a relatively small scale is that many other methods do not scale to high dimensions, as will shown in Table 5. We will show the effectiveness of our method in high dimensions later. Except for standard MTL, all the other methods need an estimate of the rank to proceed for which we apply the rank estimator in [19]. For the methods that rely on tuning parameters, we generate an independent validation set to select the tuning parameters.

We consider two coefficient matrix settings, one is only row sparse and the other one is both row sparse and column sparse. We also consider strong signal and weak signal settings. The strong signal setting is described above and for the weak signal setting, we divide the true  $\Theta^*$  by 5, resulting in a signal for which recovering true non-zero variables becomes much more difficult. Table 1 (strong signal, row sparse), Table 2 (strong signal, row and column sparse), Table 3 (weak signal, row sparse) and Table 4 (weak signal, row and column sparse) report the mean and the standard deviation of prediction errors, estimation errors and size of selected models based on 50 replications in each setting. We can see that in all the cases GDT has the lowest estimation error and prediction error. When the signal is weak, GDT may underselect the number of nonzero rows/columns, but it still has the best performance.

**Running time** We then compare the running time of all these methods. We fix a baseline model size  $n = 50, p = 80, k = 50, r = 4, s_1^* = s_2^* = 10$ , and set a free parameter  $\zeta$ . For  $\zeta = \{1, 5, 10, 20, 50, 100\}$ , each time we increase  $n, p, s_1^*, s_2^*$  by a factor of  $\zeta$  and increase  $k, r$  by a factor of  $\lfloor \sqrt{\zeta} \rfloor$  and record the running time (in seconds) of each method for a fixed tolerance level, whenever possible. We run each algorithm with a fixed reasonable tuning parameter without any validation step. If for some  $\zeta$  the algorithm does not converge in 2 hours then we simply record “>2h” and no longer increase  $\zeta$  for that method. Table 5 summarizes the results. We can see that GDT is fast even in very high dimension, while all of the other methods are computationally expensive. We note that even though GDT uses the lasso estimator in the initialization step, all the variables are

TABLE 1  
Strong signal, Row sparse

	Estimation error	Prediction error	Row support
<b>GDT</b>	$0.0452 \pm 0.0110$	$1.1060 \pm 0.0248$	$10.16 \pm 0.51$
DPP	$0.0584 \pm 0.0113$	$1.1290 \pm 0.0357$	$52.64 \pm 15.2$
TSVD	$0.3169 \pm 0.1351$	$2.4158 \pm 0.9899$	$25.62 \pm 8.03$
EEA	$0.3053 \pm 0.0998$	$1.2349 \pm 0.0362$	$84.28 \pm 6.70$
RCGL	$0.0591 \pm 0.0148$	$1.1101 \pm 0.0168$	$49.60 \pm 10.6$
JRRS	$0.0877 \pm 0.0227$	$1.1857 \pm 0.0214$	$12.26 \pm 2.02$
MTL	$0.0904 \pm 0.0243$	$1.1753 \pm 0.0204$	$73.40 \pm 2.67$

TABLE 2  
Strong signal, Row sparse and column sparse

	Estimation error	Prediction error	Row support	Column support
<b>GDT</b>	$0.0624 \pm 0.0121$	$1.0353 \pm 0.0167$	$10.24 \pm 0.65$	$10.24 \pm 0.68$
DPP	$0.0921 \pm 0.0251$	$1.0790 \pm 0.0295$	$54.10 \pm 18.25$	$10.38 \pm 0.60$
TSVD	$0.3354 \pm 0.1053$	$1.7600 \pm 0.3415$	$28.66 \pm 7.27$	$30.88 \pm 8.46$
EEA	$0.2604 \pm 0.1159$	$1.1023 \pm 0.0220$	$64.44 \pm 9.88$	$12.10 \pm 2.69$
RCGL	$0.1217 \pm 0.0325$	$1.1075 \pm 0.0174$	$42.06 \pm 7.93$	$50 \pm 0$
JRRS	$0.1682 \pm 0.0410$	$1.1612 \pm 0.0174$	$13.96 \pm 4.69$	$50 \pm 0$
MTL	$0.1837 \pm 0.0499$	$1.1652 \pm 0.0160$	$73.50 \pm 3.17$	$50 \pm 0$

TABLE 3  
Weak signal, Row sparse

	Estimation error	Prediction error	Row support
<b>GDT</b>	$0.2328 \pm 0.0474$	$1.1282 \pm 0.0231$	$10.08 \pm 0.56$
DPP	$0.2954 \pm 0.0640$	$1.1624 \pm 0.0315$	$47.26 \pm 11.7$
TSVD	$0.5842 \pm 0.1020$	$1.4271 \pm 0.0903$	$30.81 \pm 4.72$
EEA	$0.3802 \pm 0.0787$	$1.1647 \pm 0.0206$	$46.16 \pm 8.97$
RCGL	$0.2775 \pm 0.0605$	$1.1493 \pm 0.0291$	$37.92 \pm 14.4$
JRRS	$0.3600 \pm 0.0752$	$1.1975 \pm 0.0392$	$11.74 \pm 1.35$
MTL	$0.3577 \pm 0.0721$	$1.2140 \pm 0.0418$	$69.92 \pm 12.8$

TABLE 4  
Weak signal, Row sparse and column sparse

	Estimation error	Prediction error	Row support	Column support
<b>GDT</b>	$0.3173 \pm 0.0949$	$1.0380 \pm 0.0218$	$9.56 \pm 1.56$	$10.06 \pm 1.21$
DPP	$0.3899 \pm 0.0737$	$1.0580 \pm 0.0216$	$50.66 \pm 12.86$	$13.52 \pm 5.02$
TSVD	$0.6310 \pm 0.1074$	$1.1372 \pm 0.0246$	$49.94 \pm 5.53$	$43.38 \pm 2.55$
EEA	$0.6016 \pm 0.0965$	$1.0874 \pm 0.0197$	$30.64 \pm 8.65$	$30.64 \pm 8.65$
RCGL	$0.4601 \pm 0.0819$	$1.1017 \pm 0.0262$	$28.9 \pm 12.36$	$50 \pm 0$
JRRS	$0.5535 \pm 0.0866$	$1.1164 \pm 0.0262$	$12.42 \pm 6.02$	$50 \pm 0$
MTL	$0.5776 \pm 0.0873$	$1.1286 \pm 0.0296$	$53.0 \pm 18.41$	$50 \pm 0$

used in the subsequent iterations and not only the ones selected by the lasso. In particular, the speed of the method does not come from the initialization step. Table 6 summarizes the averaged number of iterations with different choices of  $\zeta$  with a tolerance value  $10^{-3}$  on the objective function. We see that the number of iteration  $T$  scales reasonably with the size of the problem.

**Effectiveness in high dimension** Next, we demonstrate the effectiveness of GDT algorithm in high dimensions. Table 1 and Table 2 are both in low

TABLE 5  
Running time comparison (in seconds)

	$\zeta = 1$	$\zeta = 5$	$\zeta = 10$	$\zeta = 20$	$\zeta = 50$	$\zeta = 100$
<b>GDT</b>	0.11	0.20	0.51	2.14	29.3	235.8
DPP	0.19	0.61	3.18	17.22	315.4	2489
TSVD	0.07	1.09	6.32	37.8	543	6075
EEA	0.50	35.6	256	>2h	>2h	>2h
RCGL	0.18	1.02	7.15	36.4	657.4	>2h
JRRS	0.19	0.82	6.36	30.0	610.2	>2h
MTL	0.18	3.12	30.92	184.3	>2h	>2h

TABLE 6  
Number of iterations in GDT algorithm with different choices of  $\zeta$

$\zeta = 1$	$\zeta = 5$	$\zeta = 10$	$\zeta = 20$	$\zeta = 50$	$\zeta = 100$
31.9	41.6	33.6	53.0	72.4	153.8

dimensions because we want to compare with other algorithms and they are slow in high dimensions, as shown in Table 5. Now we run our algorithm only and we choose  $p = 5000, k = 3000, r = 50, s_1^* = s_2^* = 100$ . The estimation error and objective value are shown in Figure 3 and Figure 4, respectively. In each figure, iteration 0 is for initialization we obtained by Lasso.

We can see that both estimation error and objective value continue to decrease, which demonstrates the effectiveness and necessity of GDT algorithm. From Figure 3 we also find that early stopping can help to avoid overfitting (although not too much), especially when  $n$  is small.

**Effect of sparsity and rank** We finally check the effect of the choices of sparsity level  $s$  and rank  $r$  on the performance of the algorithm. Here we set  $n = 100, p = 500, k = 300, r = 15, s_1^* = s_2^* = 25$ . We again consider strong signal and weak signal setting, where we divide the true  $\Theta^*$  by 5 for weak signal setting. Table 7 (strong signal, estimation error), Table 8 (strong signal, prediction error), Table 9 (strong signal, estimation error), and Table 10 (strong

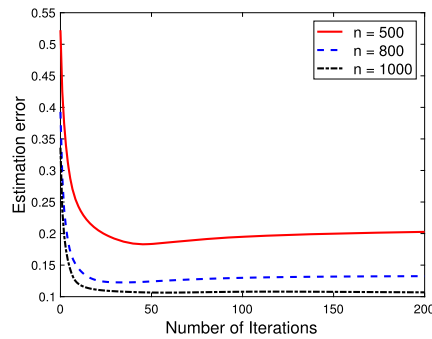


FIG 3. Estimation error

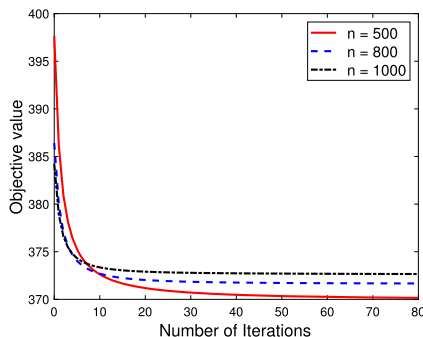


FIG 4. Objective value

TABLE 7  
 Estimation error of different choices of sparsity level and rank, with strong signal

	$r = 10$	$r = 12$	<b><math>r = 15</math></b>	$r = 20$	$r = 30$	$r = 50$	$r = 80$	$r$ selected by [19]
$s = 15$	0.6367	0.6193	0.6080	0.6228	0.6109	0.6241	0.6259	0.6164
$s = 20$	0.4752	0.4644	0.4562	0.4715	0.4532	0.4536	0.4717	0.4693
<b><math>s = 25</math></b>	0.2467	0.1668	0.0238	0.0251	0.0256	0.0261	0.0261	0.0240
$s = 30$	0.2567	0.1687	0.0288	0.0318	0.0316	0.0310	0.0314	0.0286
$s = 40$	0.2460	0.1715	0.0411	0.0424	0.0425	0.0413	0.0428	0.0437
$s = 50$	0.2495	0.1588	0.0559	0.0500	0.0518	0.0523	0.0532	0.0585
$s = 80$	0.2468	0.1725	0.1166	0.1084	0.1077	0.1095	0.1084	0.1273

signal, prediction error) report the average performance (estimation error and prediction error) of GDT algorithm with different choices of sparsity level  $s$  and rank  $r$ , based on 50 replicates in each setting. The row and column with true sparsity level and rank are highlighted as bold. In the last column, we select the rank based on the rank estimator in [19].

From the tables we see that the performance of the algorithm is poor when we underselect a sparsity level or rank. This is more significant when the signal is strong, since we are missing too many large nonzero values in the estimator. This demonstrates the necessity to be conservative when selecting sparsity level and rank. When both the sparsity level and rank are selected as greater than the true value, the algorithm performs well in a relatively large range of sparsity level and rank, especially for prediction error. As a baseline, if we estimate  $\Theta$  with Lasso estimator on each column where the regularization parameter is selected by validation set, the averaged estimation and prediction error is 0.2453 and 1.0902 for strong signal case, and 0.4716 and 1.0636 for weak signal case. Moreover, we see that overselecting a rank does not harm the performance too much, compared to overselecting a sparsity level. The rank selected by [19] performs well, and sometimes it even performs better than any of the fixed ranks.

## 5.2. Norwegian paper quality dataset

In this section we apply GDT to Norwegian paper quality dataset. This data was obtained from a controlled experiment that was carried out at a paper



TABLE 8  
Prediction error of different choices of sparsity level and rank, with strong signal

	$r = 10$	$r = 12$	$r = 15$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	$r$ selected by [19]
$s = 15$	3.6055	3.6932	3.6038	3.6659	3.6220	3.6203	3.5750	3.5658
$s = 20$	2.5595	2.5062	2.4969	2.5078	2.4935	2.4514	2.5026	2.4780
$s = 25$	1.3722	1.1432	1.0096	1.0097	1.0102	1.0103	1.0111	1.0101
$s = 30$	1.3808	1.1422	1.0129	1.0136	1.0149	1.0152	1.0146	1.0128
$s = 40$	1.3831	1.1519	1.0176	1.0208	1.0246	1.0262	1.0263	1.0196
$s = 50$	1.3966	1.1427	1.0252	1.0289	1.0371	1.0410	1.0402	1.0253
$s = 80$	1.3935	1.1650	1.0415	1.0527	1.0754	1.0969	1.1013	1.0419

TABLE 9  
Estimation error of different choices of sparsity level and rank, with weak signal

	$r = 10$	$r = 12$	$r = 15$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	$r$ selected by [19]
$s = 15$	0.6292	0.6340	0.6361	0.6313	0.6169	0.6430	0.6276	0.6203
$s = 20$	0.4733	0.4732	0.4905	0.4714	0.4747	0.4792	0.4734	0.4715
$s = 25$	0.2525	0.1877	0.1307	0.1296	0.1349	0.1365	0.1474	0.1484
$s = 30$	0.2704	0.1981	0.1474	0.1396	0.1441	0.1443	0.1364	0.1561
$s = 40$	0.2677	0.2033	0.1702	0.1772	0.1830	0.1824	0.1820	0.1810
$s = 50$	0.2695	0.2211	0.2075	0.2156	0.2281	0.2258	0.2254	0.2137
$s = 80$	0.3114	0.3008	0.2836	0.3060	0.3239	0.3474	0.3461	0.2889

factory in Norway to uncover the effect of three control variables  $X_1, X_2, X_3$  on the quality of the paper which was measured by 13 response variables. Each of the control variables  $X_i$  takes values in  $\{-1, 0, 1\}$ . To account for possible interactions and nonlinear effects, second order terms were added to the set of predictors, yielding  $X_1, X_2, X_3, X_1^2, X_2^2, X_3^2, X_1 \cdot X_2, X_1 \cdot X_3, X_2 \cdot X_3$ .

The data set can be downloaded from the website of [69] and its structure clearly indicates that dimension reduction is possible, making it a typical application for reduced rank regression methods [69, 4, 20, 111]. Based on the analysis of [19] and [4] we select the rank  $\hat{r} = 3$ ; also suggested by [19] we take  $s_1 = 6$  and  $s_2 = k = 13$  which means we have row sparsity only. GDT selects 6 of the original 9 predictors, with  $X_1^2, X_1 \cdot X_2$  and  $X_2 \cdot X_3$  discarded, which is consistent with the result in [19].

To compare prediction errors, we split the whole dataset at random, with 70% for training and 30% for test, and repeat the process 50 times to compare the performance of the above methods. All tuning parameters are selected by cross validation and we always center the responses in the training data (and transform the test data accordingly). The average RMSE on test set is shown in Table 11. We can see that GDT is competitive with the best method, demonstrating its effectiveness on real datasets.

TABLE 10  
Prediction error of different choices of sparsity level and rank, with weak signal

	$r = 10$	$r = 12$	$r = 15$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	$r$ selected by [19]
$s = 15$	1.2218	1.2173	1.2203	1.2221	1.2304	1.2265	1.2191	1.2256
$s = 20$	1.1107	1.1055	1.1076	1.1057	1.1059	1.1054	1.1050	1.1051
$s = 25$	1.0251	1.0150	1.0100	1.0096	1.0103	1.0098	1.0112	1.0110
$s = 30$	1.0291	1.0166	1.0133	1.0131	1.0142	1.0135	1.0136	1.0140
$s = 40$	1.0334	1.0235	1.0188	1.0217	1.0255	1.0267	1.0257	1.0202
$s = 50$	1.0352	1.0264	1.0247	1.0308	1.0390	1.0405	1.0376	1.0250
$s = 80$	1.0466	1.0396	1.0412	1.0547	1.0732	1.0919	1.0878	1.0390

TABLE 11  
RMSE on paper quality dataset

GDT	DPP	TSVD	EEA	RCGL	JRRS	MTL
1.002	1.012	1.094	1.161	1.001	1.013	1.014

### 5.3. Calcium imaging data

As a microscopy technique in neuroscience, calcium imaging is gaining more and more attentions [59]. It records fluorescent images from neurons and allows us to identify the spiking activity of the neurons. To achieve this goal, [104] introduces a spatiotemporal model and we briefly introduce this model here. More detailed description can be found in [104] and [93]. Denote  $k = \ell_1 \times \ell_2$  as the pixels we observe, and denote  $K$  as the total number of neurons. The observation time step is  $t = 1, \dots, T$ . Let  $S \in \mathbb{R}^{T \times K}$  be the number of spikes at each time step and for each neuron;  $A \in \mathbb{R}^{K \times k}$  be the nonnegative spatial footprint for each neuron at each pixel;  $Y \in \mathbb{R}^{T \times k}$  be the observation at each time step and at each pixel; and  $E \in \mathbb{R}^{T \times k}$  be the observation error. Ignore the baseline vector for all the pixels, the model in [104] is given by

$$Y = G^{-1}SA + E = X\Theta^* + E$$

where  $\Theta^* = SA$  is the coefficient matrix and  $X = G^{-1}$  is observed with

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -\gamma & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & -\gamma & 1 \end{pmatrix}.$$

Here  $\gamma$  is set to be  $\gamma = 1 - 1/(\text{frame rate})$  as suggested by [122]. From the settings we see that each row of  $S$  is the activation for all the neurons, and therefore it is natural to have  $S$  to be row sparse since usually we would not observe too many activations in a fixed time period; also, each column of  $A$  is the footprint for all the neurons at each pixel, and therefore it is natural to have  $A$  to be column sparse since we expect to see only a few neurons in a fixed area. Therefore our coefficient matrix  $\Theta^* = SA$  would be both row sparse and column sparse. It is also low rank since it is the product of two ‘‘tall’’ matrices because the number of neurons  $K$  are usually small. Now we see this is a multi-task learning problem with simultaneous row-sparse, column-sparse and low rank coefficient matrix where  $n = p = T$  and  $k = \ell_1 \times \ell_2$ .

We consider the calcium imaging data in [3] which is a movie with 559 frames (acquired at approximately 8.64 frames/sec), where each frame is  $135 \times 131$  pixels. This dataset is also analyzed in [93] and [59]. For this dataset, we have  $n = p = 559$  and  $k = 135 \times 131 = 17,685$ . We use  $r = 50$ , more conservative than the estimator given by [19] and we set  $s_1 = 100$  row sparsity and  $s_2 = 3000$  column sparsity. Figure 5 shows five most significant manually labeled regions; Figure 6 are the corresponding signals estimated by our GDT algorithm. We



FIG 5. Manually selected top 5 labeled regions

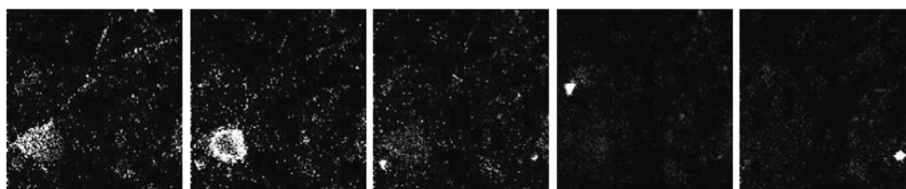


FIG 6. Corresponding signals estimated by our GDT algorithm

can see that they match very well, which demonstrates the effectiveness of our method.

## 6. Technical proofs

This section collects technical proofs.

### 6.1. Proof of Lemma 2

Let  $[\tilde{U}, \tilde{\Sigma}, \tilde{V}] = \text{rSVD}(\Theta^0)$  be the rank  $r$  SVD of the matrix  $\Theta^0$  and let

$$\tilde{\Theta} = \tilde{U}\tilde{\Sigma}(\tilde{V})^\top = \arg \min_{\text{rank}(\Theta) \leq r} \|\Theta - \Theta^0\|_F.$$

Since  $\tilde{\Theta}$  is the best rank  $r$  approximation to  $\Theta^0$ , we have

$$\|\tilde{\Theta} - \Theta^0\|_F \leq \|\Theta^0 - \Theta^*\|_F.$$

The triangle inequality gives us

$$\|\tilde{\Theta} - \Theta^*\|_F \leq \|\Theta^0 - \Theta^*\|_F + \|\Theta^0 - \tilde{\Theta}\|_F \leq 2\|\Theta^0 - \Theta^*\|_F.$$

Now that both  $\tilde{\Theta}$  and  $\Theta^*$  are rank  $r$  matrices, and according to (3.2) we have

$$\|\tilde{\Theta} - \Theta^*\|_F \leq 2\|\Theta^0 - \Theta^*\|_F \leq \frac{1}{2}\sigma_r(\Theta^*).$$

Then, Lemma 5.14 in [119] gives us

$$\begin{aligned} d^2\left(\begin{bmatrix} \tilde{U}\tilde{\Sigma}^{\frac{1}{2}} \\ \tilde{V}\tilde{\Sigma}^{\frac{1}{2}} \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\right) &\leq \frac{2}{\sqrt{2}-1} \cdot \frac{\|\tilde{\Theta} - \Theta^*\|_F^2}{\sigma_r(\Theta^*)} \\ &\leq \frac{2}{\sqrt{2}-1} \cdot \frac{4}{\sigma_r(\Theta^*)} \cdot \frac{I_0^2}{25\xi^2} \cdot \sigma_r(\Theta^*) \\ &\leq \frac{I_0^2}{\xi^2} \end{aligned}$$

where the second inequality comes from the initialization condition (3.2). Finally, Lemma 3.3 in [87] gives

$$d^2\left(\begin{bmatrix} U^0 \\ V^0 \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\right) \leq \xi^2 d^2\left(\begin{bmatrix} \tilde{U}\tilde{\Sigma}^{\frac{1}{2}} \\ \tilde{V}\tilde{\Sigma}^{\frac{1}{2}} \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\right) \leq I_0^2.$$

### 6.2. Proof of Lemma 3

For notation simplicity, let  $Z = \begin{bmatrix} U \\ V \end{bmatrix}$  denote the current iterate and let  $Z^+ = \begin{bmatrix} U^+ \\ V^+ \end{bmatrix}$  denote the next iterate. Let  $S_U = \mathcal{S}(U) \cup \mathcal{S}(U^+) \cup \mathcal{S}(U^*)$  and  $S_V = \mathcal{S}(V) \cup \mathcal{S}(V^+) \cup \mathcal{S}(V^*)$ . With some abuse of notation, we define the index set  $S_Z = S_U \cup S_V$  to represent coordinates of  $Z$  corresponding to  $U_{S_U}$  and  $V_{S_V}$ . For an index set  $S$ , let  $\mathcal{P}(U, S) = \begin{bmatrix} U_S \\ 0_{S^c} \end{bmatrix}$ . Let  $G(U, V) = f(U, V) + g(U, V)$ . Finally, let  $\Delta_U = U - U^*\hat{O}$ ,  $\Delta_V = V - V^*\hat{O}$  and  $\Delta_Z = Z - Z^*\hat{O}$ . With these notations, we can write

$$U^+ = \text{Hard}(U - \eta \cdot \nabla G_U(U, V), s_1) = \text{Hard}(U - \eta \cdot \mathcal{P}(\nabla G_U(U, V), S_U), s_1)$$

and

$$V^+ = \text{Hard}(V - \eta \cdot \nabla G_V(U, V), s_2) = \text{Hard}(V - \eta \cdot \mathcal{P}(\nabla G_V(U, V), S_V), s_2).$$

Let  $\hat{O} \in \mathcal{O}(r)$  be such that

$$d^2(Z, Z^*) = \|U - U^*\hat{O}\|_F^2 + \|V - V^*\hat{O}\|_F^2.$$

We have that

$$\begin{aligned} d^2(Z^+, Z^*) &= \min_{O \in \mathcal{O}(r)} \left\| \begin{bmatrix} U^+ \\ V^+ \end{bmatrix} - \begin{bmatrix} U^*O \\ V^*O \end{bmatrix} \right\|_F^2 \\ &\leq \left\| \begin{bmatrix} \text{Hard}(U - \eta \cdot \mathcal{P}(\nabla G_U(U, V), S_U), s_1) \\ \text{Hard}(V - \eta \cdot \mathcal{P}(\nabla G_V(U, V), S_V), s_2) \end{bmatrix} - \begin{bmatrix} U^*\hat{O} \\ V^*\hat{O} \end{bmatrix} \right\|_F^2 \\ &\leq \left(1 + \frac{2}{\sqrt{c}-1}\right) \|Z - \eta \cdot \mathcal{P}(\nabla G_Z(Z), S_Z) - Z^*\hat{O}\|_F^2, \end{aligned}$$

where the last inequality follows from Lemma 3.3 of [87]. Therefore,

$$d^2(Z^+, Z^*) \leq \left(1 + \frac{2}{\sqrt{c-1}}\right) [d^2(Z, Z^*) - 2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2)] \quad (6.1)$$

where  $T_1 = \langle \mathcal{P}(\nabla f_Z(Z), S_Z), \Delta_Z \rangle$ ,  $T_2 = \|\nabla f_Z(Z)\|_{S_Z}^2$ ,  $R_2 = \|\nabla g_Z(Z)\|_{S_Z}^2$ , and  $R_1 = \langle \mathcal{P}(\nabla g_Z(Z), S_Z), \Delta_Z \rangle$ .

For the term  $T_1$ , we have

$$\begin{aligned} T_1 &= \langle \mathcal{P}(\nabla f(UV^\top)V, S_U), \Delta_U \rangle + \langle \mathcal{P}(\nabla f(UV^\top)^\top U, S_V), \Delta_V \rangle \\ &= \underbrace{\left\langle \left[ \nabla f(UV^\top) - \nabla f(U^*V^{*\top}) \right]_{S_U, S_V}, \left[ UV^\top - U^*V^{*\top} \right]_{S_U, S_V} \right\rangle}_{T_{11}} \\ &\quad + \underbrace{\left\langle \left[ \nabla f(U^*V^{*\top}) \right]_{S_U, S_V}, \left[ UV^\top - U^*V^{*\top} \right]_{S_U, S_V} \right\rangle}_{T_{12}} \\ &\quad + \underbrace{\left\langle \left[ \nabla f(UV^\top) \right]_{S_U, S_V}, \left[ \Delta_U \Delta_V^\top \right]_{S_U, S_V} \right\rangle}_{T_{13}}. \end{aligned}$$

By restricting all the variables to the low-rank and sparse space, Theorem 2.1.11 of [101] gives

$$\begin{aligned} T_{11} &\geq \frac{L \cdot \mu}{L + \mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 \\ &\quad + \frac{1}{L + \mu} \cdot \left\| \left[ \nabla f(UV^\top) - \nabla f(U^*V^{*\top}) \right]_{S_U, S_V} \right\|_F^2 \end{aligned}$$

Next, we have

$$\begin{aligned} T_{12} &\geq - \left| \left\langle \left[ \nabla f(U^*V^{*\top}) \right]_{S_U, S_V}, \left[ UV^\top - U^*V^{*\top} \right]_{S_U, S_V} \right\rangle \right| \\ &\stackrel{(i)}{\geq} -e_{\text{stat}} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F \\ &\stackrel{(ii)}{\geq} -\frac{1}{2} \frac{L + \mu}{L \cdot \mu} e_{\text{stat}}^2 - \frac{1}{2} \frac{L \cdot \mu}{L + \mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 \end{aligned}$$

where in (i) follows from the definition of statistical error and in (ii) we used the Young's inequality  $ab \leq \frac{a^2}{2\epsilon} + \frac{\epsilon b^2}{2}$ , for  $a, b, \epsilon > 0$ . Therefore,

$$\begin{aligned} T_{11} + T_{12} &\geq \frac{1}{2} \frac{L \cdot \mu}{L + \mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 - \frac{1}{2} \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2 \\ &\quad + \frac{1}{L + \mu} \cdot \left\| \left[ \nabla f(UV^\top) - \nabla f(U^*V^{*\top}) \right]_{S_U, S_V} \right\|_F^2. \end{aligned} \quad (6.2)$$

Finally, for the term  $T_{13}$ , we have

$$\begin{aligned}
T_{13} &\geq - \left| \left\langle [\nabla f(UV^\top)]_{S_U, S_V}, [\Delta_U \Delta_V^\top]_{S_U, S_V} \right\rangle \right| \\
&\geq - \left| \left\langle [\nabla f(U^*V^{*\top})]_{S_U, S_V}, [\Delta_U \Delta_V^\top]_{S_U, S_V} \right\rangle \right| \\
&\quad - \left| \left\langle [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V}, [\Delta_U \Delta_V^\top]_{S_U, S_V} \right\rangle \right| \\
&\geq - \left( e_{\text{stat}} + \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F \right) \cdot d^2(Z, Z^*),
\end{aligned}$$

where the last inequality follows from the definition of statistical error and the observation  $\|\Delta_U \Delta_V^\top\|_F \leq \|\Delta_U\|_F \cdot \|\Delta_V\|_F \leq d^2(Z, Z^*)$ . Under the assumptions,

$$d^2(Z, Z^*) \leq \frac{4\mu_{\min}\sigma_r(\Theta^*)}{5(\mu + L)}$$

and therefore

$$\begin{aligned}
T_{13} &\geq - \left( e_{\text{stat}} + \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F \right) \\
&\quad \cdot \sqrt{\frac{4\mu_{\min}\sigma_r(\Theta^*)}{5(\mu + L)}} \cdot d(Z, Z^*) \\
&\geq - \frac{1}{2(\mu + L)} \cdot \left( e_{\text{stat}}^2 + \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F^2 \right) \\
&\quad - \frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot d^2(Z, Z^*).
\end{aligned} \tag{6.3}$$

Combining (6.2) and (6.3) we have

$$\begin{aligned}
T_1 &\geq \underbrace{\frac{1}{2} \frac{L \cdot \mu}{L + \mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2}_{T_{1a}} - \frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot d^2(Z, Z^*) \\
&\quad - \frac{1}{2} \left( \frac{L + \mu}{L \cdot \mu} + \frac{1}{L + \mu} \right) \cdot e_{\text{stat}}^2 \\
&\quad + \frac{1}{2(L + \mu)} \cdot \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F^2.
\end{aligned} \tag{6.4}$$

For the term  $T_2$ , we have

$$\begin{aligned}
\left\| [\nabla f(U^*V^{*\top})V]_{S_U} \right\|_F &= \sup_{\|U_{S_U}\|_F=1} \text{tr}(\nabla f(U^*V^{*\top})VU_{S_U}^\top) \\
&= \sup_{\|U_{S_U}\|_F=1} \langle \nabla f(U^*V^{*\top}), U_{S_U}V^\top \rangle \\
&\leq e_{\text{stat}} \cdot \|V\|_2.
\end{aligned}$$

We then have

$$\begin{aligned}
& \left\| [\nabla f(UV^\top)V]_{S_U} \right\|_F^2 \\
&= \left\| [\nabla f(UV^\top)V - \nabla f(U^*V^{*\top})V + \nabla f(U^*V^{*\top})V]_{S_U} \right\|_F^2 \\
&\leq 2 \left\| [\nabla f(UV^\top)V - \nabla f(U^*V^{*\top})V]_{S_U} \right\|_F^2 + 2 \left\| [\nabla f(U^*V^{*\top})V]_{S_U} \right\|_F^2 \\
&\leq 2 \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F^2 \cdot \|V\|_2^2 + 2e_{\text{stat}}^2 \cdot \|V\|_2^2 \\
&\leq 2 \left( \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F^2 + e_{\text{stat}}^2 \right) \cdot \|Z\|_2^2,
\end{aligned}$$

where the first inequality follows since  $\|A + B\|_F^2 \leq 2\|A\|_F^2 + 2\|B\|_F^2$ , and the last inequality follows since  $\max(\|U\|_2, \|V\|_2) \leq \|Z\|_2$ . Combining the results, we have

$$\begin{aligned}
T_2 &= \left\| [\nabla f(UV^\top)V]_{S_U} \right\|_F^2 + \left\| [\nabla f(UV^\top)^\top U]_{S_V} \right\|_F^2 \\
&\leq 4 \cdot \left( \left\| [\nabla f(UV^\top) - \nabla f(U^*V^{*\top})]_{S_U, S_V} \right\|_F^2 + e_{\text{stat}}^2 \right) \cdot \|Z\|_2^2.
\end{aligned} \tag{6.5}$$

For  $R_1$ , Lemma B.1 of [103] gives

$$\begin{aligned}
R_1 &\geq \underbrace{\frac{1}{8} \left[ \|UU^\top - U^*U^{*\top}\|_F^2 + \|VV^\top - V^*V^{*\top}\|_F^2 - 2\|UV^\top - U^*V^{*\top}\|_F^2 \right]}_{R_{12}} \\
&\quad + \underbrace{\frac{1}{2}\|\nabla g\|_F^2}_{R_{11}} - \underbrace{\frac{1}{2}\|\nabla g\|_2 \cdot \|\Delta Z\|_F^2}_{R_{13}}.
\end{aligned} \tag{6.6}$$

For  $R_{12}$ , we have that

$$\begin{aligned}
R_{12} + T_{1a} &= R_{12} + \frac{1}{8} \frac{L \cdot \mu}{L + \mu} \cdot 4 \left\| UV^\top - U^*V^{*\top} \right\|_F^2 \\
&\geq \mu_{\min} \left[ \|UU^\top - U^*U^{*\top}\|_F^2 + \|VV^\top - V^*V^{*\top}\|_F^2 \right. \\
&\quad \left. + 2\|UV^\top - U^*V^{*\top}\|_F^2 \right] \\
&= \mu_{\min} \|ZZ^\top - Z^*Z^{*\top}\|_F^2 \\
&\geq \frac{4}{5} \mu_{\min} \sigma_r^2(Z^*) \cdot d^2(Z, Z^*) \\
&= \frac{8}{5} \mu_{\min} \sigma_r(\Theta^*) \cdot d^2(Z, Z^*),
\end{aligned} \tag{6.7}$$

where the first inequality follows from the definition of  $\mu_{\min}$ , the second inequality follows from Lemma 5.4 of [119], and the last equality follows from  $\sigma_r(Z^*) = \sqrt{2}\sigma_r(\Theta^*)$ .

For  $R_{13}$ , recall that  $\Delta Z$  satisfies (3.9), we have that

$$\begin{aligned} R_{13} &\leq \frac{1}{2} \|\nabla g\|_2 \cdot \|\Delta Z\|_F \cdot \sqrt{\frac{8}{5} \mu_{\min} \sigma_r(\Theta^*)} \\ &\leq \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \cdot d^2(Z, Z^*) + \frac{1}{4} \|\nabla g\|_F^2. \end{aligned} \quad (6.8)$$

Combining (6.4), (6.6), (6.7), and (6.8), we obtain

$$\begin{aligned} T_1 + R_1 &\geq \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \cdot d^2(Z, Z^*) + \frac{1}{4} \|\nabla g\|_F^2 - \frac{1}{2} \left( \frac{L + \mu}{L \cdot \mu} + \frac{1}{L + \mu} \right) \cdot e_{\text{stat}}^2 \\ &\quad + \frac{1}{2(L + \mu)} \cdot \left\| \left[ \nabla f(UV^\top) - \nabla f(U^*V^{*\top}) \right]_{S_U, S_V} \right\|_F^2. \end{aligned} \quad (6.9)$$

For  $R_2$ , we have

$$R_2 = \|U \nabla g\|_F^2 + \|V \nabla g\|_F^2 \leq 2 \|Z\|_2^2 \cdot \|\nabla g\|_F^2. \quad (6.10)$$

Combining (6.5), (6.9), and (6.10), we have

$$\begin{aligned} &d^2(Z, Z^*) - 2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2) \\ &\leq \left( 1 - \eta \cdot \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \right) \cdot d^2(Z, Z^*) \\ &\quad + \eta \left( 4\eta \cdot \|Z\|_2^2 - \frac{1}{2(L + \mu)} \right) \cdot \left\| \left[ \nabla f(UV^\top) - \nabla f(U^*V^{*\top}) \right]_{S_U, S_V} \right\|_F^2 \\ &\quad + \eta \left( 2\eta \cdot \|Z\|_2^2 - \frac{1}{4} \right) \|\nabla g\|_F^2 \\ &\quad + \eta \left( \frac{L + \mu}{2\mu L} + \frac{1}{2(L + \mu)} + 4\eta \cdot \|Z\|_2^2 \right) \cdot e_{\text{stat}}^2. \end{aligned} \quad (6.11)$$

Under the choice of the step size,

$$\eta \leq \frac{1}{8 \|Z\|_2^2} \cdot \min \left\{ \frac{1}{2(\mu + L)}, 1 \right\},$$

the second term and third term in (6.11) are non-positive and we drop them to get

$$\begin{aligned} &d^2(Z, Z^*) - 2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2) \\ &\leq \left( 1 - \eta \cdot \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \right) \cdot d^2(Z, Z^*) + \eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2. \end{aligned} \quad (6.12)$$

Plugging (6.12) into (6.1) we finish the proof.

### 6.3. Proof of Lemma 4

Comparing (3.4) and (3.10) we see that we only need to show  $\|Z\|_2^2 \leq 2 \|Z_0\|_2^2$ . Let  $O \in \mathcal{O}(r)$  be such that

$$d^2(Z, Z^*) = \|U - U^*O\|_F^2 + \|V - V^*O\|_F^2.$$



By triangular inequality we have

$$\begin{aligned}
\|Z\|_2 &\leq \|Z^*O\|_2 + \|Z - Z^*O\|_2 \\
&\leq \|Z^*\|_2 + \sqrt{\frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot \frac{1}{\mu+L}} \\
&\leq \|Z^*\|_2 + \sqrt{\frac{4}{5} \cdot \frac{1}{8} \frac{\mu L}{\mu+L} \cdot \frac{1}{2}\sigma_r^2(Z^*) \cdot \frac{1}{\mu+L}} \quad (6.13) \\
&\leq \|Z^*\|_2 + \sqrt{\frac{1}{80}\sigma_r^2(Z^*)} \\
&\leq \frac{9}{8}\|Z^*\|_2,
\end{aligned}$$

where the third inequality follows from the definition of  $\mu_{\min}$  and  $\sigma_r^2(Z^*) = 2\sigma_r(\Theta^*)$ , and the fourth inequality follows from  $\frac{ab}{(a+b)^2} \leq \frac{1}{4}$ . Similarly, we have

$$\begin{aligned}
\|Z_0\|_2 &\geq \|Z^*O\|_2 - \|Z_0 - Z^*O\|_2 \\
&\geq \|Z^*\|_2 - \sqrt{\frac{1}{80}\sigma_r^2(Z^*)} \quad (6.14) \\
&\geq \frac{7}{8}\|Z^*\|_2.
\end{aligned}$$

Combining (6.13) and (6.14) we have

$$\|Z\|_2 \leq \frac{9}{8} \cdot \frac{8}{7} \|Z_0\|_2 \leq \sqrt{2} \|Z_0\|_2,$$

which completes the proof.

#### 6.4. Proof of Lemma 5

Let  $\Omega(s, m)$  denote a collection of subsets of  $\{1, \dots, m\}$  of size  $s$ . Let  $S_U \in \Omega(s_1, p)$  and  $S_V \in \Omega(s_2, k)$  be fixed. With some abuse of notation, let  $\mathcal{W}(S_U) = \{U \in \mathbb{R}^{p \times 2r} \mid \|U_{S_U^c}\| = 0, \|U_{S_U}\|_2 = 1\}$  and  $\mathcal{W}(S_V) = \{V \in \mathbb{R}^{k \times 2r} \mid \|V_{S_V^c}\| = 0, \|V_{S_V}\|_F = 1\}$ . Let  $\mathcal{N}_U(\epsilon)$  and  $\mathcal{N}_V(\epsilon)$  be the epsilon net of  $\mathcal{W}_U$  and  $\mathcal{W}_V$ , respectively. Using Lemma 10 and Lemma 11 of [124], we know that  $|\mathcal{N}_U(\epsilon)| \leq (3\epsilon^{-1})^{2r \cdot s_1}$ ,  $|\mathcal{N}_V(\epsilon)| \leq (3\epsilon^{-1})^{2r \cdot s_2}$ , and

$$\sup_{\substack{U \in \mathcal{W}(S_U) \\ V \in \mathcal{W}(S_V)}} \frac{1}{n} \text{tr}(E^\top XUV^\top) \leq (1-\epsilon)^{-2} \max_{\substack{U \in \mathcal{N}_U(\epsilon) \\ V \in \mathcal{N}_V(\epsilon)}} \frac{1}{n} \text{tr}(E^\top XUV^\top).$$

For fixed  $U$  and  $V$ , the random variable  $\text{tr}(E^\top XUV^\top)$  is a sub-Gaussian with variance proxy  $\sigma^2 \|X_{S_U} U_{S_U} V_{S_V}^\top\|_F^2$ . This variance proxy can be bounded as

$$\sigma^2 \|X_{S_U} U_{S_U} V_{S_V}^\top\|_F^2 \leq \sigma^2 \cdot \max_{S_U \in \Omega(s_1, p)} \|(X^\top X)_{S_U S_U}\|_2 = n\sigma^2 \bar{\kappa}(s_1).$$

Using a tail bound for sub-Gaussian random variables, we get

$$\frac{1}{n} \operatorname{tr} (E^\top X U_{S_U} V_{S_V}^\top) \leq 2\sigma \sqrt{\frac{\bar{\kappa}(s_1) \log \frac{1}{\delta}}{n}}$$

with probability at least  $1 - \delta$ . To obtain an upper bound on  $e_{\text{stat}}$ , we will apply the union bound  $\Omega(s_1, p)$ ,  $\Omega(s_2, k)$ ,  $\mathcal{N}_U(\epsilon)$  and  $\mathcal{N}_V(\epsilon)$ . We set  $\epsilon = \frac{1}{2}$  and obtain

$$e_{\text{stat}} \leq 8\sigma \sqrt{\frac{\bar{\kappa}(s_1)}{n} \left( s_1 \log p + s_2 \log k + 2r(s_1 + s_2) \log 6 + \log \frac{1}{\delta} \right)}$$

with probability at least  $1 - \delta$ . Taking  $\delta = (p \vee k)^{-1}$  completes the proof.

## 7. Conclusion

We proposed a new GDT algorithm to efficiently solve for optimization problem with simultaneous low rank and row and/or column sparsity structure on the coefficient matrix. We show the linear convergence of GDT algorithm up to statistical error. As an application, for multi-task learning problem we show that the statistical error is near optimal compared to the minimax rate. Experiments on multi-task learning demonstrate competitive performance and much faster running speed compared to existing methods. For future extensions, it would be of interest to extend GDT algorithm to non-linear models. Another potential direction would be to adaptively select the sparsity level  $s_1$  and  $s_2$  in hard thresholding step.

## Acknowledgments

This work was completed in part with resources provided by the University of Chicago Research Computing Center. We thank Zhaoran Wang and Zhuoran Yang for many useful discussions and for suggesting an application to multi-task reinforcement learning.

## References

- [1] AARONSON, S. (2007). The learnability of quantum states. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **463** 3089–3114. [MR2386653](#)
- [2] AGARWAL, A., NEGAHBAN, S. and WAINWRIGHT, M. J. (2012). Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics* **40** 1171–1197. [MR2985947](#)
- [3] AKERBOOM, J., CHEN, T. W., WARDILL, T. J., TIAN, L., MARVIN, J. S., MUTLU, S., CALDERON, N. C., ESPOSTI, F., BORGHUIS, B. G., SUN, X. R., GORDUS, A., ORGER, M. B., PORTUGUES, R., ENGERT, F., MACKLIN, J. J., FILOSA, A., AGGARWAL, A., KERR, R. A., TAKAGI, R., KRACUN, S., SHIGETOMI, E., KHAKH, B. S.,

- BAIER, H., LAGNADO, L., WANG, S. S. H., BARGMANN, C. I., KIMMEL, B. E., JAYARAMAN, V., SVOBODA, K., KIM, D. S., SCHREITER, E. R. and LOOGER, L. L. (2012). Optimization of a GCaMP calcium indicator for neural activity imaging. *Journal of Neuroscience* **32** 13819–13840.
- [4] ALDRIN, M. (1996). Moderate projection pursuit regression for multivariate response data. *Computational Statistics & Data Analysis* **21** 501–531. [MR1395234](#)
- [5] AMINI, A. A. and WAINWRIGHT, M. J. (2009). High-dimensional analysis of semidefinite relaxations for sparse principal components. *Ann. Statist.* **37** 2877–2921. [MR2541450](#)
- [6] AMIT, Y., FINK, M., SREBRO, N. and ULLMAN, S. (2007). Uncovering shared structures in multiclass classification. In *Proceedings of the 24th International Conference on Machine Learning* 17–24. ACM.
- [7] ANDO, R. K. and ZHANG, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* **6** 1817–1853. [MR2249873](#)
- [8] ARGYRIOU, A., EVGENIOU, T. and PONTIL, M. (2008). Convex multi-task feature learning. *Machine Learning* **73** 243–272.
- [9] BAHADORI, M. T., ZHENG, Z., LIU, Y. and LV, J. (2016). Scalable Interpretable Multi-Response Regression via SEED. *Technical report*.
- [10] BALAKRISHNAN, S., KOLAR, M., RINALDO, A. and SINGH, A. (2017). Recovering block-structured activations using compressive measurements. *Electron. J. Statist.* **11** 2647–2678. [MR3679905](#)
- [11] BALAKRISHNAN, S., KOLAR, M., RINALDO, A., SINGH, A. and WASSERMAN, L. (2011). Statistical and computational tradeoffs in biclustering. In *NIPS 2011 Workshop on Computational Trade-offs in Statistical Learning* **4**.
- [12] BALCAN, M., FELDMAN, V. and SZEPESVÁRI, C., eds. (2014). *Proceedings of the 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13–15, 2014. JMLR Workshop and Conference Proceedings* **35**. JMLR.org.
- [13] BARBER, R. F. and HA, W. (2017). Gradient Descent with Nonconvex Constraints: Local Concavity Determines Convergence. *Technical report*. [MR4023770](#)
- [14] BERTHET, Q. and RIGOLLET, P. (2013). Optimal detection of sparse principal components in high dimension. *Ann. Statist.* **41** 1780–1815. [MR3127849](#)
- [15] BERTSEKAS, D. P. and TSITSIKLIS, J. N. (1995). Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control* **1** 560–564 vol.1.
- [16] BHOJANAPALLI, S., KYRILLIDIS, A. and SANGHAVI, S. (2016). Dropping Convexity for Faster Semi-definite Optimization. In *29th Annual Conference on Learning Theory* (V. FELDMAN, A. RAKHLIN and O. SHAMIR, eds.). *Proceedings of Machine Learning Research* **49** 530–582. PMLR, Columbia University, New York, New York, USA.

- [17] BHOJANAPALLI, S., NEYSHABUR, B. and SREBRO, N. (2016). Global Optimality of Local Search for Low Rank Matrix Recovery. In *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, eds.) 3873–3881. Curran Associates, Inc.
- [18] BIRNBAUM, A., JOHNSTONE, I. M., NADLER, B. and PAUL, D. (2013). Minimax bounds for sparse PCA with noisy high-dimensional data. *Ann. Statist.* **41** 1055–1084. [MR3113803](#)
- [19] BUNEA, F., SHE, Y. and WEGKAMP, M. H. (2011). Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics* **39** 1282–1309. [MR2816355](#)
- [20] BUNEA, F., SHE, Y. and WEGKAMP, M. H. (2012). Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *Ann. Statist.* **40** 2359–2388. [MR3097606](#)
- [21] BURER, S. and MONTEIRO, R. D. C. (2003). A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.* **95** 329–357. Computational semidefinite and second order cone programming: the state of the art. [MR1976484](#)
- [22] BURER, S. and MONTEIRO, R. D. C. (2005). Local minima and convergence in low-rank semidefinite programming. *Math. Program.* **103** 427–444. [MR2166543](#)
- [23] CAI, J.-F., CANDÈS, E. J. and SHEN, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20** 1956–1982. [MR2600248](#)
- [24] CAI, T. T., MA, Z. and WU, Y. (2013). Sparse PCA: optimal rates and adaptive estimation. *Ann. Statist.* **41** 3074–3110. [MR3161458](#)
- [25] CAI, T. T. and ZHANG, A. (2015). ROP: matrix recovery via rank-one projections. *Ann. Statist.* **43** 102–138. [MR3285602](#)
- [26] CALANDRIELLO, D., LAZARIC, A. and RESTELLI, M. (2014). Sparse Multi-Task Reinforcement Learning. In *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger, eds.) 819–827. Curran Associates, Inc.
- [27] CANDÈS, E. J., LI, X., MA, Y. and WRIGHT, J. (2011). Robust principal component analysis? *J. ACM* **58** Art. 11, 37. [MR2811000](#)
- [28] CANDÈS, E. J. and PLAN, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE* **98** 925–936.
- [29] CANDÈS, E. J. and RECHT, B. (2009). Exact matrix completion via convex optimization. *Found. Comput. Math.* **9** 717–772. [MR2565240](#)
- [30] CANDÈS, E. J. and TAO, T. (2010). The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory* **56** 2053–2080. [MR2723472](#)
- [31] CAO, J., GU, C. and WANG, Y. (2020). Principal Component and Static Factor Analysis. In *Macroeconomic Forecasting in the Era of Big Data* 229–266. Springer.
- [32] CARUANA, R. (1997). Multitask learning. *Machine Learning* **28** 41–75.

- [33] CHANDRASEKARAN, V., SANGHAVI, S., PARRILO, P. A. and WILL-SKY, A. S. (2011). Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.* **21** 572–596. [MR2817479](#)
- [34] CHAPELLE, O., SHIVASWAMY, P., VADREUV, S., WEINBERGER, K., ZHANG, Y. and TSENG, B. (2010). Multi-Task Learning for Boosting with Application to Web Search Ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1189–1198. ACM.
- [35] CHARISOPOULOS, V., CHEN, Y., DAVIS, D., DÍAZ, M., DING, L. and DRUSVYATSKIY, D. (2019). Low-rank matrix recovery with composite optimization: good conditioning and rapid convergence. *arXiv preprint arXiv:1904.10020*.
- [36] CHEN, J., ZHOU, J. and YE, J. (2011). Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 42–50. ACM.
- [37] CHEN, K., CHAN, K.-S. and STENSETH, N. C. (2011). Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **74** 203–221. [MR2899860](#)
- [38] CHEN, L. and HUANG, J. Z. (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association* **107** 1533–1545. [MR3036414](#)
- [39] CHEN, Y. (2015). Incoherence-optimal matrix completion. *IEEE Trans. Inform. Theory* **61** 2909–2923. [MR3342311](#)
- [40] CHEN, Y., BHOJANAPALLI, S., SANGHAVI, S. and WARD, R. (2014). Coherent Matrix Completion. In *Proceedings of the 31st International Conference on Machine Learning* (E. P. XING and T. JEBARA, eds.). *Proceedings of Machine Learning Research* **32** 674–682. PMLR, Beijing, China.
- [41] CHEN, Y., JALALI, A., SANGHAVI, S. and CARAMANIS, C. (2013). Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory* **59** 4324–4337.
- [42] CHEN, Y. and WAINWRIGHT, M. J. (2015). Fast Low-Rank Estimation by Projected Gradient Descent: General Statistical and Algorithmic Guarantees. *Technical report*.
- [43] CHEN, Y.-L., KOLAR, M. and TSAY, R. S. (2019). Tensor Canonical Correlation Analysis. *arXiv preprint arXiv:1906.05358*.
- [44] COLLOBERT, R., WESTON, J., BOTTOU, L., KARLEN, M., KAVUKCUOGLU, K. and KUKSA, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12** 2493–2537.
- [45] DAVENPORT, M. A. and ROMBERG, J. (2016). An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing* **10** 608–622.
- [46] DING, L. and CHEN, Y. (2018). The leave-one-out approach for matrix

- completion: Primal and dual analysis. *arXiv preprint arXiv:1803.07554*.
- [47] ERNST, D., GEURTS, P. and WEHENKEL, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* **6** 503–556. [MR2249830](#)
- [48] EVGENIOU, T. and PONTIL, M. (2004). Regularized Multi-Task Learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 109–117. ACM.
- [49] FAN, J., DING, L., CHEN, Y. and UDELL, M. (2019). Factor Group-Sparse Regularization for Efficient Low-Rank Matrix Recovery. In *Advances in Neural Information Processing Systems* 5105–5115.
- [50] FAN, J. and LV, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B* **70** 849–911. [MR2530322](#)
- [51] FAZEL, M., HINDI, H. and BOYD, S. P. (2001). A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*. IEEE.
- [52] FAZEL, M., HINDI, H. and BOYD, S. (2004). Rank Minimization and Applications in System Theory. In *Proceedings of the 2004 American Control Conference* **4** 3273–3278.
- [53] GE, R., LEE, J. D. and MA, T. (2016). Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems* 2973–2981.
- [54] GEMULLA, R., NIJKAMP, E., HAAS, P. J. and SISMANIS, Y. (2011). Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD ’11*. ACM Press.
- [55] GROSS, D. (2011). Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Inform. Theory* **57** 1548–1566. [MR2815834](#)
- [56] GU, Q., WANG, Z. W. and LIU, H. (2016). Low-Rank and Sparse Structure Pursuit via Alternating Minimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (A. GRETTON and C. C. ROBERT, eds.)*. *Proceedings of Machine Learning Research* **51** 600–609. PMLR, Cadiz, Spain.
- [57] HA, W. and BARBER, R. F. (2017). Alternating Minimization and Alternating Descent over Nonconvex Sets. *Technical report*.
- [58] HA, W., LIU, H. and BARBER, R. F. (2018). An equivalence between stationary points for rank constraints versus low-rank factorizations. *arXiv preprint arXiv:1812.00404*.
- [59] HAEFFELE, B., YOUNG, E. and VIDAL, R. (2014). Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (T. JEBARA and E. P. XING, eds.) 2007–2015. JMLR Workshop and Conference Proceedings.
- [60] HAHN, P. R., HE, J. and LOPES, H. (2018). Bayesian factor model shrinkage for linear IV regression with many instruments. *Journal of Business*

- Economic Statistics* **36** 278–287. [MR3790214](#)
- [61] HARCHAOU, Z., DOUZE, M., PAULIN, M., DUDIK, M. and MALICK, J. (2012). Large-Scale Image Classification with Trace-Norm Regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.
- [62] HARDT, M. (2014). Understanding Alternating Minimization for Matrix Completion. In *55th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2014* 651–660. IEEE Computer Soc., Los Alamitos, CA. [MR3344916](#)
- [63] HARDT, M., MEKA, R., RAGHAVENDRA, P. and WEITZ, B. (2014). Computational Limits for Matrix Completion. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13–15, 2014* (M. BALCAN, V. FELDMAN and C. SZEPESVÁRI, eds.). *JMLR Workshop and Conference Proceedings* **35** 703–725. JMLR.org.
- [64] HARDT, M. and WOOTTERS, M. (2014). Fast Matrix Completion Without the Condition Number. In *Proceedings of the 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13–15, 2014* (M. BALCAN, V. FELDMAN and C. SZEPESVÁRI, eds.). *JMLR Workshop and Conference Proceedings* **35** 638–678. JMLR.org.
- [65] HASTIE, T., MAZUMDER, R., LEE, J. D. and ZADEH, R. (2015). Matrix completion and low-rank SVD via fast alternating least squares. *Journal of Machine Learning Research* **16** 3367–3402. [MR3450542](#)
- [66] HSIEH, C.-J. and OLSEN, P. (2014). Nuclear Norm Minimization via Active Subspace Selection. In *Proceedings of the 31st International Conference on Machine Learning* (E. P. XING and T. JEBARA, eds.). *Proceedings of Machine Learning Research* **32** 575–583. PMLR, Beijing, China.
- [67] HSU, D., KAKADE, S. M. and ZHANG, T. (2011). Robust matrix decomposition with sparse corruptions. *IEEE Trans. Inform. Theory* **57** 7221–7234. [MR2883652](#)
- [68] HUANG, J. and ZHANG, T. (2010). The benefit of group sparsity. *Annals of Statistics* **38** 1978–2004. [MR2676881](#) (2011f:62029)
- [69] IZENMAN, A. J. (2008). *Modern multivariate statistical techniques. Springer Texts in Statistics*. Springer, New York; Regression, classification, and manifold learning. [MR2445017](#)
- [70] JAIN, P., MEKA, R. and DHILLON, I. S. (2010). Guaranteed Rank Minimization via Singular Value Projection. In *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, eds.) 937–945. Curran Associates, Inc.
- [71] JAIN, P. and NETRAPALLI, P. (2015). Fast Exact Matrix Completion with Finite Samples. In *Proceedings of the 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3–6, 2015* (P. GRÜNWARD, E. HAZAN and S. KALE, eds.). *JMLR Workshop and Conference Proceedings* **40** 1007–1034. JMLR.org.
- [72] JAIN, P., NETRAPALLI, P. and SANGHAVI, S. (2013). Low-Rank Matrix Completion Using Alternating Minimization (Extended Abstract). In



- STOC'13—Proceedings of the 2013 ACM Symposium on Theory of Computing* 665–674. ACM, New York. [MR3210828](#)
- [73] JOHNSTONE, I. M. and LU, A. Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association* **104** 682–693. [MR2751448](#)
- [74] KE, Z. T. and WANG, M. (2017). A new SVD approach to optimal topic estimation. *arXiv preprint arXiv:1704.07016*.
- [75] KESHAVAN, R. H., MONTANARI, A. and OH, S. (2010). Matrix completion from a few entries. *IEEE Transactions on Information Theory* **56** 2980–2998. [MR2683452](#)
- [76] KESHAVAN, R. H., MONTANARI, A. and OH, S. (2010). Matrix completion from noisy entries. *Journal of Machine Learning Research* **11** 2057–2078. [MR2678022](#)
- [77] KIM, S. and XING, E. P. (2010). Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* 543–550.
- [78] KOLAR, M., BALAKRISHNAN, S., RINALDO, A. and SINGH, A. (2011). Minimax Localization of Structural Information in Large Noisy Matrices. In *Advances in Neural Information Processing Systems 24* (J. SHAWE-TAYLOR, R. S. ZEMEL, P. L. BARTLETT, F. C. N. PEREIRA and K. Q. WEINBERGER, eds.) 909–917.
- [79] KOLAR, M., LAFFERTY, J. D. and WASSERMAN, L. A. (2011). Union support recovery in multi-task learning. *Journal of Machine Learning Research* **12** 2415–2435. [MR2825433](#) ([2012f:62144](#))
- [80] KOLTCHINSKII, V., LOUNICI, K. and TSYBAKOV, A. B. (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Annals of Statistics* **39** 2302–2329. [MR2906869](#)
- [81] KOREN, Y., BELL, R. and VOLINSKY, C. (2009). Matrix factorization techniques for recommender systems. *Computer* **42** 30–37.
- [82] LAPIN, M., SCHIELE, B. and HEIN, M. (2014). Scalable Multitask Representation Learning for Scene Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1434–1441.
- [83] LAZARIC, A. and GHAVAMZADEH, M. (2010). Bayesian Multi-Task Reinforcement Learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21–24, 2010, Haifa, Israel* (J. FÜRNKRANZ and T. JOACHIMS, eds.) 599–606. Omnipress.
- [84] LEE, K. and BRESLER, Y. (2010). ADMiRA: atomic decomposition for minimum rank approximation. *IEEE Trans. Inform. Theory* **56** 4402–4416. [MR2807337](#)
- [85] LEE, M., SHEN, H., HUANG, J. Z. and MARRON, J. S. (2010). Biclustering via sparse singular value decomposition. *Biometrics* **66** 1087–1095. [MR2758496](#)
- [86] LI, Q., ZHU, Z. and TANG, G. (2017). Geometry of Factored Nuclear Norm Regularization. *Technical report*.
- [87] LI, X., ZHAO, T., ARORA, R., LIU, H. and HAUPT, J. (2016). Stochastic Variance Reduced Optimization for Nonconvex Sparse Learning. In *Pro-*



- ceedings of the 33rd International Conference on International Conference on Machine Learning – Volume 48. ICML'16* 917–925. JMLR.org.
- [88] LIU, H. and BARBER, R. F. (2018). Between hard and soft thresholding: optimal iterative thresholding algorithms. *arXiv preprint arXiv:1804.08841*.
- [89] LIU, Z. and VANDENBERGHE, L. (2009). Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.* **31** 1235–1256. [MR2558821](#)
- [90] LOUNICI, K., PONTIL, M., TSYBAKOV, A. B. and VAN DE GEER, S. A. (2011). Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics* **39** 2164–204. [MR2893865](#)
- [91] MA, X., XIAO, L. and WONG, W. H. (2014). Learning regulatory programs by threshold SVD regression. *Proceedings of the National Academy of Sciences* **111** 15675–15680.
- [92] MA, Z. (2013). Sparse principal component analysis and iterative thresholding. *Ann. Statist.* **41** 772–801. [MR3099121](#)
- [93] MA, Z., MA, Z. and SUN, T. (2014). Adaptive Estimation in Two-way Sparse Reduced-rank Regression. *Technical report*.
- [94] MEI, S., BAI, Y., MONTANARI, A. et al. (2018). The landscape of empirical risk for nonconvex losses. *The Annals of Statistics* **46** 2747–2774. [MR3851754](#)
- [95] MEI, S., CAO, B. and SUN, J. (2012). Encoding Low-Rank and Sparse Structures Simultaneously in Multi-Task Learning. Technical Report, Microsoft Technical Report.
- [96] MUNOS, R. and SZEPESVÁRI, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research* **9** 815–857. [MR2417255](#)
- [97] NA, S., KOLAR, M. and KOYEJO, O. (2019). Estimating differential latent variable graphical models with applications to brain connectivity. *arXiv preprint arXiv:1909.05892*.
- [98] NEGAHBAN, S. and WAINWRIGHT, M. J. (2011). Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics* **39** 1069–1097. [MR2816348](#)
- [99] NEGAHBAN, S. and WAINWRIGHT, M. J. (2012). Restricted strong convexity and weighted matrix completion: optimal bounds with noise. *Journal of Machine Learning Research* **13** 1665–1697. [MR2930649](#)
- [100] NEGAHBAN, S. N., RAVIKUMAR, P., WAINWRIGHT, M. J. and YU, B. (2012). A unified framework for high-dimensional analysis of  $M$ -estimators with decomposable regularizers. *Statistical Science* **27** 538–557. [MR3025133](#)
- [101] NESTEROV, Y. (2013). *Introductory Lectures on Convex Optimization*. Springer US. [MR3839649](#)
- [102] OBOZINSKI, G., WAINWRIGHT, M. J. and JORDAN, M. I. (2011). Support union recovery in high-dimensional multivariate regression. *Annals of Statistics* **39** 1–47. [MR2797839](#)
- [103] PARK, D., KYRILLIDIS, A., CARAMANIS, C. and SANGHAVI, S. (2016).

- Finding low-rank solutions to matrix problems, efficiently and provably. *arXiv preprint arXiv:1606.03168*. [MR3860118](#)
- [104] PNEVMATIKAKIS, E. A., GAO, Y., SOUDRY, D., PFAU, D., LACEFIELD, C., POSKANZER, K., BRUNO, R., YUSTE, R. and PANINSKI, L. (2014). A Structured Matrix Factorization Framework for Large Scale Calcium Imaging Data Analysis. *Technical report*.
- [105] RECHT, B. (2011). A simpler approach to matrix completion. *Journal of Machine Learning Research* **12** 3413–3430. [MR2877360](#)
- [106] RECHT, B., FAZEL, M. and PARRILO, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **52** 471–501. [MR2680543](#)
- [107] RECHT, B. and RÉ, C. (2013). Parallel stochastic gradient algorithms for large-scale matrix completion. *Math. Program. Comput.* **5** 201–226. [MR3069879](#)
- [108] RICHARD, E., ANDRE SAVALLE, P. and VAYATIS, N. (2012). Estimation of Simultaneously Sparse and Low Rank Matrices. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)* (J. LANGFORD and J. PINEAU, eds.) 1351–1358. ACM, New York, NY, USA.
- [109] ROHDE, A. and TSYBAKOV, A. B. (2011). Estimation of high-dimensional low-rank matrices. *Ann. Statist.* **39** 887–930. [MR2816342](#)
- [110] SELTZER, M. L. and DROPPA, J. (2013). Multi-Task Learning in Deep Neural Networks for Improved Phoneme Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013* 6965–6969. IEEE.
- [111] SHE, Y. (2017). Selective factor extraction in high dimensions. *Biometrika* **104** 97–110. [MR3626471](#)
- [112] SHE, Y. and TRAN, H. (2019). On cross-validation for sparse reduced rank regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **81** 145–161. [MR3904783](#)
- [113] SNEL, M. and WHITESON, S. (2012). Multi-Task Reinforcement Learning: Shaping and Feature Selection. In *Lecture Notes in Computer Science 237–248*. Springer, Berlin, Heidelberg.
- [114] SREBRO, N., RENNIE, J. and JAAKKOLA, T. S. (2005). Maximum-Margin Matrix Factorization. In *Advances in Neural Information Processing Systems 17* (L. K. Saul, Y. Weiss and L. Bottou, eds.) 1329–1336. MIT Press.
- [115] SUN, R. and LUO, Z.-Q. (2016). Guaranteed matrix completion via non-convex factorization. *IEEE Trans. Inform. Theory* **62** 6535–6579. [MR3565131](#)
- [116] SUTTON, R. S. and BARTO, A. G. (1998). *Introduction to Reinforcement Learning*, 1st ed. MIT Press, Cambridge, MA, USA. [MR3889951](#)
- [117] TAKÁCS, G., PILÁSZY, I., NÉMETH, B. and TIKK, D. (2007). Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter* **9** 80.
- [118] TIBSHIRANI, R. J. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* **58** 267–288. [MR1379242 \(96j:62134\)](#)

- [119] TU, S., BOCZAR, R., SIMCHOWITZ, M., SOLTANOLKOTABI, M. and RECHT, B. (2016). Low-Rank Solutions of Linear Matrix Equations via Procrustes Flow. In *Proceedings of The 33rd International Conference on Machine Learning* (M. F. BALCAN and K. Q. WEINBERGER, eds.). *Proceedings of Machine Learning Research* **48** 964–973. PMLR, New York, New York, USA.
- [120] TURLACH, B. A., VENABLES, W. N. and WRIGHT, S. J. (2005). Simultaneous variable selection. *Technometrics* **47** 349–363. [MR2164706](#)
- [121] UEMATSU, Y., FAN, Y., CHEN, K., LV, J. and LIN, W. (2017). SOFAR: Large-Scale Association Network Learning. *Technical report*. [MR3988531](#)
- [122] VOGELSTEIN, J. T., PACKER, A. M., MACHADO, T. A., SIPPY, T., BABADI, B., YUSTE, R. and PANINSKI, L. (2010). Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology* **104** 3691–3704.
- [123] VOUNOU, M., JANOUSOVA, E., WOLZ, R., STEIN, J. L., THOMPSON, P. M., RUECKERT, D. and MONTANA, G. (2012). Sparse reduced-rank regression detects genetic associations with voxel-wise longitudinal phenotypes in Alzheimer’s disease. *NeuroImage* **60** 700–716.
- [124] VU, V. (2011). Singular vectors under random perturbation. *Random Struct. Alg.* **39** 526–538. [MR2846302](#)
- [125] VU, V. Q. and LEI, J. (2013). Minimax sparse principal subspace estimation in high dimensions. *Ann. Statist.* **41** 2905–2947. [MR3161452](#)
- [126] WANG, J., KOLAR, M. and SREBRO, N. (2016). Distributed Multi-Task Learning with Shared Representation. *Technical report*.
- [127] WANG, J., KOLAR, M. and SREBRO, N. (2016). Distributed Multi-Task Learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (A. GRETTON and C. C. ROBERT, eds.). *Proceedings of Machine Learning Research* **51** 751–760. PMLR, Cadiz, Spain.
- [128] WANG, Z., LAI, M.-J., LU, Z., FAN, W., DAVULCU, H. and YE, J. (2015). Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM J. Sci. Comput.* **37** A488–A514. [MR3313832](#)
- [129] WANG, Z., LU, H. and LIU, H. (2014). Nonconvex Statistical Optimization: Minimax-Optimal Sparse PCA in Polynomial Time. *Technical report*.
- [130] WEINBERGER, K., DASGUPTA, A., LANGFORD, J., SMOLA, A. and ATTENBERG, J. (2009). Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* 1113–1120. ACM.
- [131] WILSON, A., FERN, A., RAY, S. and TADEPALLI, P. (2007). Multi-task reinforcement learning. In *Proceedings of the 24th international conference on Machine learning – ICML ’07*. ACM Press.
- [132] XIANG, S., ZHU, Y., SHEN, X. and YE, J. (2012). Optimal Exact Least Squares Rank Minimization. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD ’12*. ACM Press.

- [133] XU, P., MA, J. and GU, Q. (2017). Speeding Up Latent Variable Gaussian Graphical Model Estimation via Nonconvex Optimizations. *Technical report*.
- [134] XUE, Y., LIAO, X., CARIN, L. and KRISHNAPURAM, B. (2007). Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research* **8** 35–63. [MR2280214](#)
- [135] YAN, Q., YE, J. and SHEN, X. (2015). Simultaneous pursuit of sparseness and rank structures for matrix decomposition. *Journal of Machine Learning Research* **16** 47–75. [MR3317229](#)
- [136] YANG, D., MA, Z. and BUJA, A. (2014). A sparse singular value decomposition method for high-dimensional data. *J. Comput. Graph. Statist.* **23** 923–942. [MR3270704](#)
- [137] YU, M., GUPTA, V. and KOLAR, M. (2016). Statistical Inference for Pairwise Graphical Models Using Score Matching. In *Advances in Neural Information Processing Systems* **29**. Curran Associates, Inc.
- [138] YU, M., GUPTA, V. and KOLAR, M. (2017). An Influence-Receptivity Model for Topic Based Information Cascades. *Technical report*.
- [139] YU, M., THOMPSON, A. M., RAMAMURTHY, K. N., YANG, E. and LOZANO, A. C. (2017). Multitask Learning using Task Clustering with Applications to Predictive Modeling and GWAS of Plant Varieties. *Technical report*.
- [140] YUAN, M., EKICI, A., LU, Z. and MONTEIRO, R. (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B* **69** 329–346. [MR2323756](#)
- [141] YUAN, M. and LIN, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B* **68** 49–67. [MR2212574](#)
- [142] YUAN, X.-T. and ZHANG, T. (2013). Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research* **14** 899–925. [MR3063614](#)
- [143] ZHANG, X., WANG, L. and GU, Q. (2017). A Nonconvex Free Lunch for Low-Rank plus Sparse Matrix Recovery. *Technical report*.
- [144] ZHAO, T., WANG, Z. and LIU, H. (2015). Nonconvex low rank matrix factorization via inexact first order oracle. *Advances in Neural Information Processing Systems*.
- [145] ZHENG, Q. and LAFFERTY, J. D. (2015). A Convergent Gradient Descent Algorithm for Rank Minimization and Semidefinite Programming from Random Linear Measurements. In *Advances in Neural Information Processing Systems* **28** (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, eds.) 109–117. Curran Associates, Inc.
- [146] ZHOU, J., LIU, J., NARAYAN, V. A. and YE, J. (2013). Modeling disease progression via multi-task learning. *NeuroImage* **78** 233–248.
- [147] ZHU, Y., SHEN, X. and YE, C. (2016). Personalized prediction and sparsity pursuit in latent factor models. *J. Amer. Statist. Assoc.* **111** 241–252. [MR3494656](#)
- [148] ZHU, Z., LI, Q., TANG, G. and WAKIN, M. B. (2017). Global Optimality

- in Low-rank Matrix Optimization. *Technical report*. [MR3832394](#)
- [149] ZHU, Z., LI, Q., TANG, G. and WAKIN, M. B. (2017). The Global Optimization Geometry of Nonsymmetric Matrix Factorization and Sensing. *Technical report*.
- [150] ZHUANG, Y., CHIN, W.-S., JUAN, Y.-C. and LIN, C.-J. (2013). A Fast Parallel SGD for Matrix Factorization in Shared Memory Systems. In *Proceedings of the 7th ACM Conference on Recommender Systems – RecSys '13*. ACM Press.