Statistica Sinica Preprint No: SS-2024-0423

Title	Simultaneous Estimation and Dataset Selection for Transfer Learning in High Dimensions by a Non-convex Penalty
Manuscript ID	SS-2024-0423
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202024.0423
Complete List of Authors	Zeyu Li, Dong Liu, Yong He and Xinsheng Zhang
Corresponding Authors	Dong Liu
E-mails	liudong_stat@163.com

Notice: Accepted author version.

Statistica Sinica

Simultaneous Estimation and Dataset Selection for Transfer Learning in High Dimensions by the Truncated Norm Penalty

Zeyu Li, Dong Liu, Yong He and Xinsheng Zhang

Southeast University, Nanyang Technological University,

Shandong University and Fudan University

Abstract: In this paper, we introduce a method for simultaneous parameter estimation and informative source dataset identification in high-dimensional transfer learning, leveraging the truncated norm penalty function. This integrated approach contrasts with conventional strategies that treat useful dataset selection and transfer learning as separate steps. To solve the resulting non-convex optimization problem, specifically under sparse linear regression and generalized low-rank trace regression models, we adopt the difference of convex (DC) programming with the alternating direction method of multipliers (ADMM) procedure. We theoretically justify the proposed algorithm from both statistical and computational perspectives. Numerical results are reported alongside to validate the theoretical assertions. An `R` package `MHDTL` is developed to implement the

Dong Liu is the corresponding author, email: liudong_stat@163.com.

The `R` package is available at `https://github.com/heyongstat/MHDTL`.

proposed methods.

Key words and phrases: clustering analysis; DC-ADMM; knowledge transfer; M-estimators.

1. Introduction

Transfer learning, a concept originating from the computer science community (Torrey and Shavlik, 2010; Zhuang et al., 2020; Niu et al., 2020), has been applied to various *high-dimensional* statistical problems in recent years (Bastani, 2021; Li et al., 2022; Tian and Feng, 2023; Cai and Wei, 2021). As its name indicates, useful information from related tasks (*sources*) could be transferred to the original task (*target*) to improve the efficiency of statistical inference for the latter.

Under the high-dimensional statistics setting, where the number of parameters can be much larger than the sample sizes, additional low-dimensional structures are often imposed on the model parameters to avoid the curse of dimensionality and derive consistent estimators. With the intrinsic low-dimensionality assumption, flourishing regularized (or penalized) convex optimization methods have been proposed to achieve statistically optimal performance within polynomial time. One typical example is the sparse signal assumption, i.e., assuming the sparsity of the model parameters of interest (Tibshirani, 1996; Fan and Li, 2001; Candes and Tao, 2007), which often involves the lasso-type penalty. When the parameters of interest arise in matrix form, an alternative model assumption is low rank, which has been widely explored and applied in the fields of statistics, computer science, and econometrics (Zhou and Li, 2014; Fan et al., 2019; He et al., 2023). In this article, we follow Negahban et al. (2012) and work under the high-dimensional M-estimator framework, which includes sparse linear regression and generalized low-rank trace regression models as concrete examples.

1.1 Literature review and contributions

Intuitively, when the sources are sufficiently similar to the target, transfer learning could outperform the direct estimation procedure that uses only the target dataset. Specifically, let's consider the ideal case when all source distributions are identical to the target distribution; in that case, the statistically optimal estimation procedure is to first pool all the datasets and make a global pooling estimation. However, the knowledge transfer estimators might sometimes exhibit much poorer performance than the single-task learning estimator using the target dataset alone, a phenomenon known as *negative transfer* (Torrey and Shavlik, 2010). This naturally leads to the following question: *what might be the reason for negative transfer in the* *high-dimensional statistical context?*

The fundamental reason is that the data distributions of the target and sources are generally not identical. It is typically summarized in the following two points in the high-dimensional supervised transfer learning literature (He et al., 2024): 1. *model shift*, which often refers to the difference in model parameters, e.g., the regression coefficients; 2. *covariate shift*, which is often related to the Hessian matrices of the population loss functions, e.g., the population covariance matrices of the covariates in the case of linear regression.

In Li et al. (2022); Tian and Feng (2023), the authors quantified the model shift using a specific distance measure for the target and source parameters. We also adopt the distance similarity in this article: if the source parameter vector is sufficiently close to the target one, we say the source is useful (or informative) for the target task, or that the model shift is slight. When the informative sources are known in advance, which is also called the oracle transfer setting in the literature, Bastani (2021); Li et al. (2022); Tian and Feng (2023) proposed using a two-step procedure: in the first step, an oracle pooling estimator is acquired by pooling the useful datasets; and in the second step, this estimator is debiased using only the target data. Meanwhile, in the computer science literature, the idea of fitting a shared global model for all related local clients even appears in personalized federated learning, e.g., the `FEDAVG` algorithm (McMahan et al., 2017), when the heterogeneity of data distributions is kept in mind. To some extent, `FEDAVG` targets the oracle pooling estimator. Indeed, in Theorem 3.3 of Chen et al. (2021), the authors show that the minimax optimal rate in knowledge transfer is achieved by either single-task learning or data pooling estimation under the distance similarity in low dimensions.

In high-dimensional cases, the previously mentioned two-step methods are often sensitive to the covariate shift, even if the model shift is slight for all sources. Hence, these methods often require additional homogeneity conditions on the Hessian matrices of the population loss functions. Quite recently, Li et al. (2024); He et al. (2024) provided sophisticated knowledge transfer methods that take both slight model shifts and covariate shifts into account. Enlightened by the claims of Chen et al. (2021) concerning low dimensional cases, we show that the statistical performance of the high-dimensional oracle pooling estimator is still guaranteed by simply enlarging the regularization parameter from the default rate, regardless of the covariate shift. Specific statistical models are discussed within our framework, including generalized low-rank trace regression, whose knowledge transfer problem, to the best of our knowledge, has not been addressed in the existing literature.

Unfortunately, the useful sources with small model shifts are generally unknown in practice. That is to say, there might be useless and even potentially harmful sources, with generally larger model shifts included for consideration. To address this problem and acquire the oracle pooling estimator in the non-oracle setting, Li et al. (2022, 2024) took advantage of model selection aggregation, while Tian and Feng (2023) proposed a data-driven method to first detect the transferable sources in a separate step before applying their oracle transfer learning algorithm. However, these methods often require solving optimization problems repeatedly with the entire source dataset, which results in an additional computational burden and a loss of data privacy.

In this work, we introduce the truncated norm penalty, which is a non-convex penalty that originated from the clustering analysis literature (Shen et al., 2012; Pan et al., 2013; Wu et al., 2016; Liu et al., 2023). This enables the automatic incorporation of those highly informative sources (with a slight model shift) into the estimation procedure of the target task while ignoring the non-informative ones (with a larger model shift). In contrast to existing methods that focus on identifying useful datasets as a separate step, our algorithm directly outputs the estimator by simultaneously selecting the useful sources. Moreover, the DC-ADMM algorithm in this article only requires the parameters from the sources during the iterations rather than the entire datasets. This is particularly important if preserving data privacy and saving communication costs are also kept in mind (McMahan et al., 2017).

In summary, the contributions of this article are stated as follows. We propose introducing the truncated norm penalty to estimate the target parameters while simultaneously selecting the useful sources. A difference of convex (DC) programming with the alternating direction method of multipliers (ADMM) algorithm is used to solve the resulting non-convex optimization problem concerning two specific statistical models, including generalized low-rank trace regression. The knowledge transfer problem of these models has not been discussed in the literature, as far as we know. The proposed algorithm is theoretically justified from both statistical and computational perspectives. An `R` package, `MHDTL`, has been developed that is also used in the numerical experiments to support our claims.

1.2 Framework

We consider the transfer learning problem under the high-dimensional M-estimator framework with decomposable regularizers, following Negahban et al. (2012). We denote the target distribution as $\mathbb{D}_0$ and K potential source distributions as $\mathbb{D}_k$ and $k = 1, \cdots, K$, whose population parameters are denoted as $\boldsymbol{\theta}_k^* \in \mathbb{R}^p$ and $k \in \{0\} \cup [K]$. We assume that $\boldsymbol{\theta}_k^*$ is the minimizer of the expected loss function, i.e.,

$$\boldsymbol{\theta}_k^* = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathbb{E}\ [\mathcal{L}_k(\boldsymbol{Z}_k; \boldsymbol{\theta})], \quad \boldsymbol{Z}_k \text{ is sampled from } \mathbb{D}_k,$$

where $\mathcal{L}_k(\cdot; \boldsymbol{\theta})$ is the loss function of the k-th study. In this work, we mainly focus on high-dimensional regression problems, and K is treated as a finite constant. For instance, we can let the k-th dataset consist of n_k *i.i.d.* observations of $\boldsymbol{Z}_{k,i} = (\boldsymbol{X}_{k,i}, y_{k,i})$, set $y_{k,i} = \langle \boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,i} \rangle + \varepsilon_{k,i}$, where $\varepsilon_{k,i}$ are standard Gaussian errors. Then, $\boldsymbol{\theta}_k^*$ would be the minimizer of the expected loss function $\mathbb{E}(\|y_k - \langle \boldsymbol{\theta}, \boldsymbol{X}_k \rangle\|^2)$. For another example, one may consider the generalized linear model $\mathbb{P}(y_{k,i} | \boldsymbol{X}_{k,i}) \propto \exp\{y_{k,i} \langle \boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,i} \rangle - b(\langle \boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,i} \rangle)\}$, where $\boldsymbol{\theta}_k^* = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathbb{E}\{-y_k \langle \boldsymbol{\theta}, \boldsymbol{X}_k \rangle + b(\langle \boldsymbol{\theta}, \boldsymbol{X}_k \rangle)\}$.

In the high-dimensional statistical context, we assume that $\boldsymbol{\theta}_0^*$ lies in a low-dimensional subspace $\mathcal{M} \subset \mathbb{R}^p$. For instance, it might be the subspace of vectors with a particular support or the subspace of low-rank matrices. Denote the inner product induced norm as $\|\cdot\|^2 = \langle \cdot, \cdot \rangle$. Given the subspace $\overline{\mathcal{M}}$ such that $\mathcal{M} \subset \overline{\mathcal{M}}$, let $\overline{\mathcal{M}}^\perp = \{\boldsymbol{v} \in \mathbb{R}^p \,|\, \langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0 \text{ for all } \boldsymbol{u} \in \overline{\mathcal{M}}\}$ be the orthogonal complement of the space $\overline{\mathcal{M}}$. We say a norm-based regularizer (or penalty) $\mathcal{R}$ is decomposable with respect to $(\mathcal{M}, \overline{\mathcal{M}}^{\perp})$ if

$$\mathcal{R}(\boldsymbol{\theta}+\boldsymbol{\gamma}) = \mathcal{R}(\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\gamma}), \quad \text{for all } \boldsymbol{\theta} \in \mathcal{M} \text{ and } \boldsymbol{\gamma} \in \overline{\mathcal{M}}^{\perp}.$$

For the sake of better illustration, we present here several examples of the decomposable regularizers given in Negahban et al. (2012).

Example 1 (Sparse vector and ℓ_1 norm)**.** Suppose that $\boldsymbol{\theta}_0^*$ is supported on a subset $S \subset \{0, 1, \ldots, p\}$ with cardinality s, we set $\mathcal{M} = \overline{\mathcal{M}} = \{\boldsymbol{\theta} \in \mathbb{R}^p \mid \boldsymbol{\theta}_j = 0 \text{ for all } j \notin S\}$. We can take $\mathcal{R}(\cdot) = \|\cdot\|_1$, it is easy to verify that $\|\boldsymbol{\theta}+\boldsymbol{\gamma}\|_1 = \|\boldsymbol{\theta}\|_1 + \|\boldsymbol{\gamma}\|_1$ for all $\boldsymbol{\theta} \in \mathcal{M}$ and $\boldsymbol{\gamma} \in \overline{\mathcal{M}}^{\perp}$ by the construction of the subspaces. If we consider the linear regression model, we face the transfer learning problem for sparse linear regression discussed in Li et al. (2022). If we take the generalized linear model instead, the problem degenerates to the one discussed in Tian and Feng (2023).

Example 2 (Low-rank matrix and nuclear norm)**.** Let $\boldsymbol{\theta}_0^*$ be a low-rank matrix, and let $\boldsymbol{\theta}_0^* = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}$ be its singular value decomposition (SVD), where $\boldsymbol{D}$ is a diagonal matrix consisting of non-increasing singular values. Denote the first r columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ by $\boldsymbol{U}^r$ and $\boldsymbol{V}^r$, we take

$$\boldsymbol{\theta}_0^* \in \mathcal{M} := \left\{\boldsymbol{\theta} \in \mathbb{R}^{d_1 \times d_2} \mid \operatorname{row}(\boldsymbol{\theta}) \subseteq \operatorname{col}\left(\mathbf{V}^r\right), \operatorname{col}(\boldsymbol{\theta}) \subseteq \operatorname{col}\left(\mathbf{U}^r\right)\right\}, \tag{1.1}$$

$$\overline{\mathcal{M}}^{\perp} := \left\{\boldsymbol{\theta} \in \mathbb{R}^{d_1 \times d_2} \mid \operatorname{row}(\boldsymbol{\theta}) \perp \operatorname{col}\left(\mathbf{V}^r\right), \operatorname{col}(\boldsymbol{\theta}) \perp \operatorname{col}\left(\mathbf{U}^r\right)\right\}, \tag{1.2}$$

where col$(\cdot)$ and row$(\cdot)$ denote spaces spanned by columns and rows. It is known that the matrix nuclear norm $\|\cdot\|_N$ satisfies the decomposability condition. Note that here $\mathcal{M}$ is not equal to $\overline{\mathcal{M}}$. Similarly, given the linear regression model, we can study transfer learning for low-rank trace regression. The low-rank trace regression is introduced in Zhou and Li (2014) and is particularly useful in modeling matrix completion, multi-task learning, and compressed sensing problems (Hamidi and Bayati, 2022). Meanwhile, for the generalized linear model, we have generalized low-rank trace regression with various applications including generalized reduced-rank regression and one-bit matrix completion (Fan et al., 2019).

Finally, the characterization of similarity between the target and the sources is essential for us to design algorithms that fully utilize the shared information and provide a sound theoretical analysis. As previously alluded to, we consider the following popular distance similarity: for the informative sources in a subset $\mathcal{A} \subseteq \{1, \cdots K\}$, we control the magnitudes of model shift by providing an upper bound on the distance between the difference vectors. Namely, for $\boldsymbol{\delta}_k^* = \boldsymbol{\theta}_0^* - \boldsymbol{\theta}_k^*$, we assume for some norm $\mathcal{B}_k$ that

$$\mathcal{B}_k(\boldsymbol{\delta}_k^*) \leq h, \quad k \in \mathcal{A}, \tag{1.3}$$

where smaller h indicates a higher similarity. For example, $\mathcal{B}_k$ can be the vector ℓ_1 or the ℓ_2 norm in the case of sparse linear regression. For generalized trace regression, it can be the nuclear norm, the Frobenius norm, or the vectorized ℓ_1 norm. As for $k \in \mathcal{A}^c$, $\boldsymbol{\theta}_k^*$ is allowed to be quite different from $\boldsymbol{\theta}_0^*$. Meanwhile, we make no assumptions about covariate similarity; i.e., there is no additional homogeneity condition on the Hessian matrices of the population loss functions that is required in (Li et al., 2022; Tian and Feng, 2023).

1.3 Organization and notations

The remainder of this paper is organized as follows. In Section 2, we introduce the truncated norm penalty and the resulting non-convex optimization problem. We then provide the computational details for solving the non-convex problem under the sparse linear regression and generalized low-rank trace regression settings. In Section 3, we discuss the statistical properties of the proposed algorithm. We report numerical simulation results in Section 4. Finally, the proposed algorithm is applied to two real datasets in Section 5. In the supplementary material, we provide theoretical arguments concerning the two specific statistical models and remark on the optional fine-tuning step. Moreover, we present additional numerical details that further support our arguments. Finally, we provide the proofs of the theoretical results.

To end this section, we state some commonly used notations. The inner product induced norm $\|\cdot\|$ is sometimes written as $\|\cdot\|_2$ for vectors or $\|\cdot\|_F$ for matrices. For a real vector $\boldsymbol{a}$, let $\|\boldsymbol{a}\|_1$ be its ℓ_1-norm, while for a real matrix $\boldsymbol{A}$, let $\|\boldsymbol{A}\|_{op}$ and $\|\boldsymbol{A}\|_N$ be its operator norm (from ℓ_2 to ℓ_2) and nuclear norm, respectively. We use the standard O_p notation for stochastic boundedness. For a sub-Gaussian random variable X, we define the sub-Gaussian norm by $\|X\|_{\Psi_2} = \inf\{t > 0 : \mathbb{E}\exp(X^2/t^2) \le 2\}$; while for a random vector $\boldsymbol{X}$, its sub-Gaussian norm is defined as $\|\boldsymbol{X}\|_{\Psi_2} = \sup_{\|\boldsymbol{x}\|_2=1} \|\langle \boldsymbol{X}, \boldsymbol{x}\rangle\|_{\Psi_2}$. In the end, we write $x \lesssim y$ if $x \le Cy$ for some $C > 0$, $x \gtrsim y$ if $x \ge cy$ for some $c > 0$, and $x \asymp y$ if both $x \lesssim y$ and $x \gtrsim y$ hold. Note that the constants C and c may not be identical in different lines.

2. Methodology

In the oracle case, we know which of the sources are informative; namely, we know the subset $\mathcal{A}$ of small model shifts in advance. Let $\mathcal{P} := \{0\} \cup \mathcal{A}$, for $n_{\mathcal{P}} := \sum_{k\in\mathcal{P}} n_k$, we have the oracle pooling estimator:

$$\widehat{\boldsymbol{\theta}}_{\mathcal{P}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{n_{\mathcal{P}}} \sum_{k\in\mathcal{P}} \sum_{i\le n_k} \mathcal{L}_k(\boldsymbol{Z}_{k,i};\boldsymbol{\theta}) + \lambda_{\mathcal{P}}\mathcal{R}(\boldsymbol{\theta}). \tag{2.4}$$

When the informative sources are unknown, in order to eliminate the influence of non-informative or even harmful sources, we propose a truncated norm penalized algorithm based on the intuition of dataset clustering. For each study $k \in \{0\} \cup [K]$, with the population parameter $\boldsymbol{\theta}_k^*$, let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_0, \cdots, \boldsymbol{\theta}_K)$, $\widehat{\boldsymbol{\Theta}} = (\widehat{\boldsymbol{\theta}}_0, \cdots, \widehat{\boldsymbol{\theta}}_K) \in \mathbb{R}^{p \times (K+1)}$,

$$\begin{aligned} \widehat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta} \in \mathbb{R}^{p \times (K+1)}}{\operatorname{argmin}} \frac{1}{N} \sum_{k=0}^{K} \sum_{i=1}^{n_k} \mathcal{L}_k\left(\boldsymbol{Z}_{k,i}; \boldsymbol{\theta}_k\right) + \sum_{k=0}^{K} \frac{n_k \lambda_{\mathcal{P}}}{N} \mathcal{R}\left(\boldsymbol{\theta}_k\right) \\ + \sum_{k=1}^{K} \frac{n_k \lambda_{\mathcal{Q}_k}}{N} \mathcal{Q}\left(\boldsymbol{\theta}_k - \boldsymbol{\theta}_0\right). \end{aligned} \tag{2.5}$$

Here $N = \sum_{k=0}^{K} n_k$, while $\mathcal{Q}(\cdot) = \min\left[\mathcal{R}(\cdot), \tau\right]$ is the truncated norm penalty for the tuning parameter $\tau > 0$. The first column of $\widehat{\boldsymbol{\Theta}}$, namely $\widehat{\boldsymbol{\theta}}_0$, is set to be the estimator of $\boldsymbol{\theta}_0^*$.

Intuitively, (2.5) can be viewed as data pooling in a slacker manner. Rather than directly enforcing $\boldsymbol{\theta}_k = \boldsymbol{\theta}_0$, we penalize the distance between them using the penalty function $\mathcal{Q}$, which possesses the additional truncating nature to cut off the influence of $\boldsymbol{\theta}_k$ if it is too far away from $\boldsymbol{\theta}_0$. As discussed extensively in Duan and Wang (2023), such slackness often leads to robustness.

Specifically, the tuning parameters $\lambda_{\mathcal{Q}_k}$ and τ control the strength and range of dataset clustering, respectively. If $\lambda_{\mathcal{Q}_k} = 0$ or $\tau = 0$, then (2.5) calculates each $\widehat{\boldsymbol{\theta}}_k$ using the k-th dataset alone, with a penalty $\mathcal{R}$ and a tuning parameter $\lambda_{\mathcal{P}}$. On the other hand, if $\lambda_{\mathcal{Q}_k} \to \infty$ and $\tau \to \infty$, then the optimization problem (2.5) is equivalent to the optimization problem (2.4) that blind-pools all the sources and the target dataset.

If $\lambda_{\mathcal{P}}$ and all $\lambda_{\mathcal{Q}_k}$ are fixed, but only τ tends to infinity, then the problem will turn into a (numerically stable) convex problem that blindly includes all source datasets. In fact, if the individual contrast $\boldsymbol{\delta}_k^* = \boldsymbol{\theta}_0^* - \boldsymbol{\theta}_k^*$ has a certain low-dimensional structure (e.g., sparsity), the resulting convex problem will be in the same spirit as the one proposed by Gross and Tibshirani (2016); Ollier and Viallon (2017); Li et al. (2024), such that the target parameter $\boldsymbol{\theta}_0^*$ and the contrast vectors $\boldsymbol{\delta}_k^*$ from the sources are estimated simultaneously. On the other hand, if $\boldsymbol{\delta}_k^*$ has no such low-dimensional structure but is still small as measured by $\mathcal{B}_k$, the decomposable regularizer $\mathcal{R}$ in $\mathcal{Q}$ tends to penalize $\widehat{\boldsymbol{\theta}}_0 - \widehat{\boldsymbol{\theta}}_k$ towards zero.

In the end, the dataset selection capability of (2.5) stems from the fact that: those non-informative datasets identified by $\mathcal{R}(\widehat{\boldsymbol{\theta}}_0 - \widehat{\boldsymbol{\theta}}_k) > \tau$ would have no influence on the estimator $\widehat{\boldsymbol{\theta}}_0$ due to the truncation in $\mathcal{Q}$. Similarly, the estimation of $\widehat{\boldsymbol{\theta}}_k$ is also independent of other datasets if $\mathcal{R}(\widehat{\boldsymbol{\theta}}_0 - \widehat{\boldsymbol{\theta}}_k) > \tau$ for the same reason. Hence, we also penalize $\boldsymbol{\theta}_k$ by $\mathcal{R}(\cdot)$ to ensure numerical stability in case we have to estimate the high-dimensional vector $\boldsymbol{\theta}_k^* \in \mathbb{R}^p$ using the k-th dataset alone, which is slightly different from the methods in Gross and Tibshirani (2016); Ollier and Viallon (2017); Li et al. (2024), where only $\boldsymbol{\theta}_0$ and $\boldsymbol{\delta}_k^*$ are penalized.

2.1 Computational aspects

To numerically solve the resulting non-convex problem (2.5), we adopt the difference of convex (DC) programming with the alternating direction method of multipliers (ADMM) procedures (Boyd et al., 2011; Wu et al., 2016; Fan et al., 2019, 2021). In this article, we work on two specific statistical models under (2.5), namely sparse linear regression and generalized low-rank trace regression.

2.1.1 Sparse linear regression

We first present the sparse linear regression case, where $\sum_{i \le n_k} \mathcal{L}_k\left(\boldsymbol{Z}_{k,i}; \boldsymbol{\theta}_k\right) = \|\boldsymbol{y}_k - \mathcal{X}_k \boldsymbol{\theta}_k\|_2^2$ and $\mathcal{R} = \|\cdot\|_1$. For $\boldsymbol{\delta}_k = \boldsymbol{\theta}_0 - \boldsymbol{\theta}_k$, we focus on the rescaled problem $\widehat{\boldsymbol{\Theta}} = \operatorname{argmin}_{\boldsymbol{\Theta},\boldsymbol{\delta}} S(\boldsymbol{\Theta}, \boldsymbol{\delta})$ where

$$S(\Theta, \boldsymbol{\delta}) = \sum_{k=0}^{K} \alpha_k \|\boldsymbol{y}_k - \mathcal{X}_k \boldsymbol{\theta}_k\|_2^2 + \sum_{k=0}^{K} n_k \lambda_{\mathcal{P}} \|\boldsymbol{\theta}_k\|_1 + \sum_{k=1}^{K} n_k \lambda_{\mathcal{Q}_k} \min(\|\boldsymbol{\delta}_k\|_1, \tau),$$

$$\text{subject to} \quad \boldsymbol{\delta}_k = \boldsymbol{\theta}_0 - \boldsymbol{\theta}_k, \quad 1 \le k \le K.$$

Note that the objective function $S(\Theta, \boldsymbol{\delta})$ is non-convex, which poses substantial challenges for optimization. To address this issue, we first employ the difference of-convex (DC) programming framework (Thi Hoai An and Dinh Tao, 1997), which reformulates the objective as the difference of two convex functions: $S(\boldsymbol{\Theta}, \boldsymbol{\delta}) = S_1(\boldsymbol{\Theta}, \boldsymbol{\delta}) - S_2(\boldsymbol{\delta})$ where $S_1(\boldsymbol{\Theta}, \boldsymbol{\delta}) =$

$\sum_{k=0}^{K} \alpha_k \|\boldsymbol{y}_k - \mathcal{X}_k \boldsymbol{\theta}_k\|_2^2 + \sum_{k=0}^{K} n_k \lambda_{\mathcal{P}} \|\boldsymbol{\theta}_k\|_1 + \sum_{k=1}^{K} n_k \lambda_{\mathcal{Q}_k} \|\boldsymbol{\delta}_k\|_1$ and $S_2(\boldsymbol{\delta}) = \sum_{k=1}^{K} n_k \lambda_{\mathcal{Q}_k} (\|\boldsymbol{\delta}_k\|_1 - \tau)_+$, with $a_+ = \max(a, 0)$ for any $a \in \mathbb{R}$. To handle the non-convexity in $-S_2(\boldsymbol{\delta})$, we adopt its piecewise affine minorization (Wu et al., 2016):

$$S_2^{(m+1)}(\boldsymbol{\delta}) = S_2(\widehat{\boldsymbol{\delta}}^{(m)}) + \sum_{k=1}^{K} n_k \lambda_{\mathcal{Q}_k} \left(\|\boldsymbol{\delta}_k\|_1 - \|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_1 \right) I \left(\|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_1 \geq \tau \right),$$

where $\widehat{\boldsymbol{\delta}}^{(m)}$, to be formally defined in (2.6), denotes the estimate at the m-th iteration.

Actually, this construction arises from a first-order Taylor expansion of $S_2(\boldsymbol{\delta})$ at the differentiable point $||\widehat{\boldsymbol{\delta}}^{(m)}||_1$ and serves as a standard step in DC programming; see Thi Hoai An and Dinh Tao (1997), Wu et al. (2016), and Liu et al. (2023) for details. By combining these definitions with some straightforward algebra, one can verify that $S_2^{(m+1)}(\boldsymbol{\delta})$ provides a valid sequence of lower approximations, as $S_2^{(m+1)}(\boldsymbol{\delta}) \leq S_2(\boldsymbol{\delta})$ always holds. Moreover, $S_2^{(m+1)}(\boldsymbol{\delta})$ naturally coincides with the surrogate function required in the Majorization–Minimization (MM) framework (Sun et al., 2016), whose main idea is to replace the original optimization problem with a sequence of surrogate functions that are easier to optimize. Motivated by this principle, we substitute $S_2(\boldsymbol{\delta})$ with its lower approximation $S_2^{(m+1)}(\boldsymbol{\delta})$ in the m-th iteration, which yields the following upper approximation of

$S(\Theta, \boldsymbol{\delta})$:

$$S^{(m+1)}(\boldsymbol{\Theta}, \boldsymbol{\delta}) = \sum_{k=0}^{K} \alpha_k \|\boldsymbol{y}_k - \mathcal{X}_k \boldsymbol{\theta}_k\|_2^2 + \sum_{k=0}^{K} n_k \lambda_{\mathcal{P}} \|\boldsymbol{\theta}_k\|_1$$
$$+ \sum_{k=1}^{K} n_k \lambda_{\mathcal{Q}_k} \left[\|\boldsymbol{\delta}_k\|_1 I\left(\|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_1 < \tau \right) + \tau I\left(\|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_1 \geq \tau \right) \right],$$

Accordingly, at the m-th iteration, we consider the following optimization:

$$\left(\widehat{\boldsymbol{\Theta}}^{(m+1)}, \widehat{\boldsymbol{\delta}}^{(m+1)}\right) = \underset{\boldsymbol{\Theta}, \boldsymbol{\delta}}{\operatorname{argmin}} \ S^{(m+1)}(\boldsymbol{\Theta}, \boldsymbol{\delta}), \text{ subject to } \boldsymbol{\delta}_k = \boldsymbol{\theta}_0 - \boldsymbol{\theta}_k. \quad (2.6)$$

It follows that $S^{(m+1)}(\boldsymbol{\Theta}, \boldsymbol{\delta})$ is convex in both $\boldsymbol{\Theta}$ and $\boldsymbol{\delta}$ under the equality constraint, which naturally motivates the use of the ADMM algorithm to obtain the global minimizer of $S^{(m+1)}(\boldsymbol{\Theta}, \boldsymbol{\delta})$ for each $m \geq 0$. These procedures are standard, and we leave the implementation details to the supplementary material to save space. Finally, the following proposition guaranties the numerical convergence of the DC-ADMM algorithm within a finite number of steps.

Proposition 1 (Convergence of DC-ADMM)**.** *The DC-ADMM algorithm in this section converges in finite steps; namely, there exists $m^* < \infty$ such that $S(\widehat{\boldsymbol{\Theta}}^{(m)}, \widehat{\boldsymbol{\delta}}^{(m)}) = S(\widehat{\boldsymbol{\Theta}}^{(m^*)}, \widehat{\boldsymbol{\delta}}^{(m^*)})$ for $m \geq m^*$. Moreover, $(\widehat{\boldsymbol{\Theta}}^{(m^*)}, \widehat{\boldsymbol{\delta}}^{(m^*)})$ is a Karush-Kuhn-Tucker (KKT) point.*

2.1.2 Generalized low-rank trace regression

We briefly introduce the procedure to solve the optimization problem in equation (2.5) under the case with $\mathcal{L}_k(\boldsymbol{Z}_{k,i};\boldsymbol{\theta}_k) = -y_{k,i}\eta_{k,i} + b(\eta_{k,i})$ for $\eta_{k,i} = \langle \boldsymbol{\theta}_k, \boldsymbol{X}_{k,i}\rangle$ and $\mathcal{R} = \|\cdot\|_N$. In line with the DC-ADMM procedures presented in Section 2.1, the difference of convex procedure enables us to focus on a sequence of upper approximations. However, optimizing arbitrary loss functions with a nuclear norm penalty is still a challenging problem. We can replace $\boldsymbol{L}_k(\boldsymbol{\theta}) = \sum_{1\le i\le n_k} \mathcal{L}_k(\boldsymbol{Z}_{k,i};\boldsymbol{\theta})/n_k$ with its quadratic approximation via the iterative Peaceman-Rachford splitting method, following Fan et al. (2019, 2021), simplifying the optimization problem to

$$
\begin{aligned}
\left(\widehat{\Theta}^{(m+1)}, \widehat{\boldsymbol{\delta}}^{(m+1)}, \widehat{\gamma}\right) &= \underset{\Theta,\boldsymbol{\delta},\boldsymbol{\gamma}}{\operatorname{argmin}}\ S^{(m+1)}(\Theta, \boldsymbol{\delta}, \boldsymbol{\gamma}) \\
\text{subject to}\quad & \boldsymbol{\delta}_k = \boldsymbol{\theta}_0 - \boldsymbol{\theta}_k,\ 1 \le k \le K, \\
& \boldsymbol{\gamma}_k = \boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k^{(m)},\ 0 \le k \le K,
\end{aligned}
$$

for

$$
\begin{aligned}
S^{(m+1)}(\boldsymbol{\Theta}, \boldsymbol{\delta}, \boldsymbol{\gamma}) = & \sum_{k=0}^{K} n_k\alpha_k \boldsymbol{Q}(\boldsymbol{\gamma}_k; \widehat{\boldsymbol{\theta}}_k^{(m)}) + \sum_{k=0}^{K} n_k\lambda_{\mathcal{P}}\|\boldsymbol{\theta}_k\|_N \\
& + \sum_{k=1}^{K} n_k\lambda_{\mathcal{Q}_k}\left[\|\boldsymbol{\delta}_k\|_N I(\|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_N < \tau) + \tau I(\|\widehat{\boldsymbol{\delta}}_k^{(m)}\|_N \ge \tau)\right],
\end{aligned}
\tag{2.7}
$$

$$
\boldsymbol{Q}(\boldsymbol{\gamma}_k; \widehat{\boldsymbol{\theta}}_k^{(m)}) = \operatorname{vec}^\top(\boldsymbol{\gamma}_k)\,\nabla^2 \boldsymbol{L}_k(\widehat{\boldsymbol{\theta}}_k^{(m)})\operatorname{vec}(\boldsymbol{\gamma}_k)/2 + \operatorname{vec}^\top(\boldsymbol{\gamma}_k)\operatorname{vec}\left[\nabla \boldsymbol{L}_k\left(\widehat{\boldsymbol{\theta}}_k^{(m)}\right)\right].
$$

Combined with Theorem 2.1 of Cai et al. (2010) and Boyd et al. (2011), we can implement standard ADMM procedures to solve the above optimization problem (2.7) with a nuclear norm penalty. The rest is analogous to the sparse linear regression case, and we also leave the details to the supplementary material.

Remark 1 (Default initialization and a shrinking τ strategy)**.** As one can imagine, initialization is vital for the performance of the proposed non-convex optimization: non-convexity entitles the algorithm to dataset selection capability but at the cost of numerical stability.

In the default implementations of our numerical experiments and also the `R` package `MHDTL`, the initial estimates are taken as the local estimates using standard high-dimensional methods, e.g., performing lasso using the k-th dataset alone. The default initial estimates are generally accurate, and they can be theoretically justified by statistical theory. However, in extreme cases where the initialization is particularly poor, e.g., when the local target estimator is close to the estimators of the non-informative sources, it might lead to a vicious cycle in which the target estimation is influenced by potentially harmful information.

Inspired by the intuition that the problem becomes "more convex" as $\tau \to \infty$ (see the discussion right after (2.5)), we suggest a shrinking τ

The `R` package is available at `https://github.com/heyongstat/MHDTL`.

strategy to enhance numerical stability and avoid the "vicious cycle" in many cases, even under very poor initializations. The key idea is to iteratively take the output of our algorithm equipped with a larger τ as the input for our algorithm equipped with a smaller τ. By iteratively shrinking τ to a suitable value (i.e., small enough to separate the useless sources from the target estimation but large enough to include the useful sources; cross-validation can be used to evaluate performance here), we are able to alleviate the impact of poor initialization.

In the numerical experiments, the shrinking τ strategy is able to ensure the numerical stability of the non-convex algorithm, even with very poor initializations. We leave detailed discussions and additional numerical results to the supplementary material.

3. Theory

This section is devoted to the statistical properties of the given method. We begin with the oracle setting, where the informative subset $\mathcal{A}$ is known in advance. For $\mathcal{P} = \{0\} \cup \mathcal{A}$, recall that the oracle pooling estimator is

$$\widehat{\boldsymbol{\theta}}_{\mathcal{P}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{n_{\mathcal{P}}} \sum_{k\in\mathcal{P}} \sum_{i\leq n_k} \mathcal{L}_k(\boldsymbol{Z}_{k,i}; \boldsymbol{\theta}) + \lambda_{\mathcal{P}} \mathcal{R}(\boldsymbol{\theta}).$$

We introduce some useful notations from Negahban et al. (2012). Define $\boldsymbol{L}_k(\boldsymbol{\theta}) = \sum_{1\leq i\leq n_k} \mathcal{L}_k(\boldsymbol{Z}_{k,i}; \boldsymbol{\theta})/n_k$, $k = 0, \cdots K$; then the optimization problem (2.4) can be rewritten as $\widehat{\boldsymbol{\theta}}_{\mathcal{P}} = \mathrm{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}) + \lambda_{\mathcal{P}}\mathcal{R}(\boldsymbol{\theta})$ for $\boldsymbol{L}_{\mathcal{P}} = \sum_{k\in\mathcal{P}} n_k\boldsymbol{L}_k/n_{\mathcal{P}}$. Recall that the regularizer $\mathcal{R}$ is decomposable with respect to $(\mathcal{M},\overline{\mathcal{M}}^{\perp})$ such that $\boldsymbol{\theta}_0^* \in \mathcal{M} \subset \overline{\mathcal{M}}$. Let $\mathcal{R}^*(\boldsymbol{v}) = \sup_{\mathcal{R}(\boldsymbol{u})\le 1}\langle \boldsymbol{u},\boldsymbol{v}\rangle$ be the dual norm of $\mathcal{R}$. Define

$$\delta\boldsymbol{L}(\boldsymbol{\Delta};\boldsymbol{\theta}_0^*) = \boldsymbol{L}(\boldsymbol{\theta}_0^* + \boldsymbol{\Delta}) - \boldsymbol{L}(\boldsymbol{\theta}_0^*) - \langle \nabla\boldsymbol{L}(\boldsymbol{\theta}_0^*),\boldsymbol{\Delta}\rangle\,,$$

and $\psi(\mathcal{M}) = \sup_{\boldsymbol{u}\in\mathcal{M}\setminus\{0\}}\mathcal{R}(\boldsymbol{u})/\|\boldsymbol{u}\|$ is the subspace compatibility constant. Throughout this work, we shall take the view that $\psi(\mathcal{M}) < \infty$ for simplicity, namely, the dimension of $\mathcal{M}$ does not diverge as $p\to\infty$. In the end, define the cone-like set

$$\mathbb{C}\left(\mathcal{M},\overline{\mathcal{M}}^{\perp};\boldsymbol{\theta}^*\right) := \left\{\boldsymbol{\Delta}\in\mathbb{R}^p \mid \mathcal{R}\left(\boldsymbol{\Delta}_{\overline{\mathcal{M}}^{\perp}}\right) \le 3\mathcal{R}\left(\boldsymbol{\Delta}_{\overline{\mathcal{M}}}\right) + 4\mathcal{R}(\boldsymbol{\theta}^*_{\mathcal{M}^{\perp}})\right\},$$

where the subscript, e.g., $\boldsymbol{\Delta}_{\overline{\mathcal{M}}}$ means the projection of $\boldsymbol{\Delta}$ onto the subspace $\overline{\mathcal{M}}$. In the classical fixed p large n regime, the convergence of M-estimators often requires the notion of strong convexity or, equivalently, a strictly positive Hessian matrix in a neighborhood of the population minimizer (Van der Vaart, 2000). However, in the high-dimensional setting that $p \gg n$, the $p\times p$ Hessian matrix will always be rank-deficient, even for linear regression models (Negahban et al., 2012). Hence, we need to relax strong convexity by restricting the set of directions to the cone-like set $\mathbb{C}(\mathcal{M},\overline{\mathcal{M}}^{\perp};\boldsymbol{\theta}_0^*)$. The following proposition follows directly from Theorem 1 of Negahban et al.

(2012).

Proposition 2. *Suppose $\boldsymbol{L}_{\mathcal{P}}$ is convex, differentiable, and satisfies the restricted strong convexity (RSC) condition:*

$$\delta \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\Delta};\boldsymbol{\theta}_0^*) \geq \kappa_{\mathcal{P}}\|\boldsymbol{\Delta}\|^2 - \tau_{\mathcal{P}}, \quad \textit{for all } \boldsymbol{\Delta} \in \mathbb{C}\left(\mathcal{M},\overline{\mathcal{M}}^{\perp};\boldsymbol{\theta}_0^*\right), \tag{3.8}$$

with $\tau_{\mathcal{P}} \lesssim \lambda_{\mathcal{P}}\psi^2(\overline{\mathcal{M}})/\kappa_{\mathcal{P}}$. Solving the problem (2.4) with $\lambda_{\mathcal{P}} \geq 2\mathcal{R}^*\left(\nabla \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}_0^*)\right)$, we have

$$\left\|\widehat{\boldsymbol{\theta}}_{\mathcal{P}} - \boldsymbol{\theta}_0^*\right\|^2 \lesssim \frac{\lambda_{\mathcal{P}}^2}{\kappa_{\mathcal{P}}^2}\psi^2(\overline{\mathcal{M}}), \quad \mathcal{R}\left(\widehat{\boldsymbol{\theta}}_{\mathcal{P}} - \boldsymbol{\theta}_0^*\right) \lesssim \frac{\lambda_{\mathcal{P}}}{\kappa_{\mathcal{P}}}\psi^2(\overline{\mathcal{M}}). \tag{3.9}$$

As remarked in Negahban et al. (2012), the arguments here are actually deterministic statements about the convex program (2.4). When we deal with particular statistical models, we need to calculate $\mathcal{R}^*[\nabla \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}_0^*)]$ and verify the RSC condition on a case by case basis via probabilistic analysis.

Now, we take a closer look at $\lambda_{\mathcal{P}}$ in the convergence rate of (3.9). As it is required that $\lambda_{\mathcal{P}} \geq 2\mathcal{R}^*[\nabla \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}_0^*)]$, we focus on the right hand side. Assume that all $\boldsymbol{L}_k$ have the second-order derivative at $\boldsymbol{\theta}_k^*$. let $\boldsymbol{\delta}_k^* = \boldsymbol{\theta}_0^* - \boldsymbol{\theta}_k^*$, by Taylor expansion and the triangular inequality,

$$\begin{aligned}\mathcal{R}^*\left(\nabla \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}_0^*)\right) &= \mathcal{R}^*\left(\sum_{k\in\mathcal{P}} n_k\left[\nabla \boldsymbol{L}_k(\boldsymbol{\theta}_k^*) + \nabla^2 \boldsymbol{L}_k(\boldsymbol{\theta}_0^*)\boldsymbol{\delta}_k^* + \boldsymbol{r}_k(\boldsymbol{\delta}_k^*)\right]\right)/n_{\mathcal{P}} \\ &\le \underbrace{\mathcal{R}^*\left(\sum_{k\in\mathcal{P}} n_k \nabla \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)\right)/n_{\mathcal{P}}}_{v_{\mathcal{P}}} + \underbrace{\sum_{k\in\mathcal{P}} n_k \mathcal{R}^*\left(\nabla^2 \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)\boldsymbol{\delta}_k^*\right)/n_{\mathcal{P}}}_{b_{\mathcal{P}}} \\ &\quad + \underbrace{\mathcal{R}^*\left(\sum_{k\in\mathcal{P}} n_k \boldsymbol{r}_k(\boldsymbol{\delta}_k^*)\right)/n_{\mathcal{P}}}_{\text{remainder}}.\end{aligned} \tag{3.10}$$

Since $\boldsymbol{\theta}_k^*$ are assumed to be the minimizers of population loss functions, which means $\mathbb{E}[\nabla \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)] = 0$. The first term $v_{\mathcal{P}}$ could be viewed as the variance term and is often well-controlled by standard high-dimensional probabilistic arguments, proportional to $\sqrt{n_{\mathcal{P}}}$. The second term $b_{\mathcal{P}}$ could be viewed as the bias term and is the source of the regularization enlargement. Define the $(\mathcal{B}_k, \mathcal{R}^*)$-operator norm of the $m \times n$ matrix $\boldsymbol{A}$ by $\|\boldsymbol{A}\|_{\mathcal{B}_k\to\mathcal{R}^*} = \sup_{\mathcal{B}_k(\boldsymbol{v})\le 1} \mathcal{R}^*(\boldsymbol{A}\boldsymbol{v})$. We have $b_{\mathcal{P}} \lesssim h$ as long as $\mathcal{B}_k(\boldsymbol{\delta}_k^*) \le h$ and $\|\nabla^2 \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)\|_{\mathcal{B}_k\to\mathcal{R}^*}$ can be bounded above by some constants. Finally, for sufficiently small contrast, as measured by h, the remainder term could be absorbed into the second term.

To better illustrate the effect of enlarging regularization, we further extend the above arguments towards the two detailed statistical models introduced earlier, namely the sparse linear regression and the generalized low-rank trace regression. We leave these extensional arguments for the supplementary material.

Finally, with the previous analysis of the oracle pooling estimator in hand, we are now able to theoretically justify our method of solving the non-convex optimization problem in (2.5). Indeed, we can show that the oracle pooling estimator $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$ is a local minimum of (2.5) under some mild conditions.

Theorem 1 (Oracle local minimum)**.** *Let $\mathcal{A}$ denote the informative source datasets, and $\mathcal{A}^c$ denote the rest. For $k \in \mathcal{A}$, assume that $\mathcal{B}_k(\boldsymbol{\delta}_k^*) \leq h$ and*

$$\max(\|\nabla^2 \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)\|_{\mathcal{R}\to\mathcal{R}^*}, \|\nabla^2 \boldsymbol{L}_k(\boldsymbol{\theta}_k^*)\|_{\mathcal{B}_k\to\mathcal{R}^*}) \leq M.$$

In addition, assume that the conditions in Proposition 2 hold for $\mathcal{P} = \{0\} \cup \mathcal{A}$. As for $k \in \mathcal{A}^c$, define $\widehat{\boldsymbol{\theta}}'_k = \operatorname{argmin}_{\boldsymbol{\theta}_k \in \mathbb{R}^p} \boldsymbol{L}_k(\boldsymbol{\theta}_k) + \lambda_{\mathcal{P}} \mathcal{R}(\boldsymbol{\theta}_k)$, and assume that $\mathcal{R}(\widehat{\boldsymbol{\theta}}'_k - \boldsymbol{\theta}_0^*) > 2\tau$ for some $\tau > 0$ (the same τ as in $\mathcal{Q}$). Denote $v_k = \mathcal{R}^*(\nabla \boldsymbol{L}_k(\boldsymbol{\theta}_k^*))$ for the optimization problem (2.5) with $\lambda_{\mathcal{P}} \gtrsim \mathcal{R}^*(\nabla \boldsymbol{L}_{\mathcal{P}}(\boldsymbol{\theta}_0^*)) \asymp (v_{\mathcal{P}} + h)$ and $\lambda_{\mathcal{Q}_k} \gtrsim (v_k + h)$ as $\min_{k\in\mathcal{A}} n_k \to \infty$, $p \to \infty$ with $\max_{k\in\mathcal{A}} v_k < \infty$, $v_{\mathcal{P}} \to 0$, and $h \to 0$. there exists a local minimum $\widehat{\boldsymbol{\Theta}}$ of (2.5) whose first column satisfies $\widehat{\boldsymbol{\theta}}_0 = \widehat{\boldsymbol{\theta}}_{\mathcal{P}}$.

Theorem 1, together with Proposition 1, justifies the performance of the DC-ADMM algorithm from both statistical and computational aspects. In some cases, such as FEDAVG (McMahan et al., 2017), the oracle pooling estimator $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$ is applied to all local clients. That is to say, $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$ itself could be directly used as an estimator of $\boldsymbol{\theta}_0^*$, and its statistical properties are guaranteed by our theoretical analysis. Meanwhile, for the two-step methods, an additional fine-tuning step using the target dataset is frequently utilized. That is to say, we can choose to fine-tune the primal estimator $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$ using the target dataset by solving

$$\widehat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n_0} \sum_{i\leq n_0} \mathcal{L}_0 \left(\boldsymbol{Z}_{0,i}; \widehat{\boldsymbol{\theta}}_{\mathcal{P}} + \boldsymbol{\delta} \right) + \lambda_d \mathcal{R}\left(\boldsymbol{\delta}\right),$$

and setting $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}^{\star} := \widehat{\boldsymbol{\theta}}_{\mathcal{P}} + \widehat{\boldsymbol{\delta}}$. In this work, the additional superscript $\cdot^{\star}$ is used to represent the fine-tuned version of any knowledge transfer estimator. In the supplementary material, we briefly remark on the optional fine-tuning step after acquiring the initial estimator.

4. Simulations

In this section, we report the numerical performance of the proposed method based on simulation studies. In the sparse linear regression case, we compare the statistical performance of our truncated-penalized method with some state-of-the-art methods in the literature. Meanwhile, for the generalized low-rank trace regression, to the best of our knowledge, the problem of knowledge transfer remains unaddressed in the literature. Hence, in this case, we compare our algorithm with some ad hoc methods available. The results of the generalized low-rank trace regression are reported in the supplementary material.

For the sparse linear regression, we generate data from the linear model $y_{k,i} = \langle \boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,i} \rangle + \varepsilon_{k,i}$ for $i \in [n_k]$ and $k \in \{0\} \cup [10]$, where $\varepsilon_{k,i}$ is drawn independently from $N(0,1)$. For $\boldsymbol{X}_{k,i}$, we consider the following cases: (a) homogeneous covariates: draw $\boldsymbol{X}_{k,i}$ from $N(\mathbf{0}, \boldsymbol{I}_p)$ independently; (b) heterogeneous covariates: let $\boldsymbol{\Lambda}_k$ be a matrix of $1.5p$ rows and p columns, whose elements are drawn independently from $N(0,1)$; then, draw $\boldsymbol{X}_{k,i}$ independently from $N(\mathbf{0}, \boldsymbol{\Sigma}_k)$ with $\boldsymbol{\Sigma}_k = 2\boldsymbol{\Lambda}_k^\top \boldsymbol{\Lambda}_k/(3p)$. We set $n_0 = 250$, $p = 500$, and $n_k = 400$ for $k \neq 0$. Further numerical results are provided in the supplementary material, which show that the proposed method adapts readily to higher dimensions and more correlated covariates.

As for $\boldsymbol{\theta}_k^*$ and $k \in \mathcal{P} := \{0, \ldots, 5\}$, we consider two configurations. We set $\boldsymbol{\theta}_{0j}^* = 0.4$ for $j \in [s] = \{1, \cdots, s\}$, where $\boldsymbol{\theta}_{kj}^*$ represents the j-th element of $\boldsymbol{\theta}_k^*$. Then we consider: (a) sparse contrasts: for each $k \in [5]$, let H_k be a random subset of $[p]$ with $|H_k| = 3$ and let $\boldsymbol{\theta}_{kj}^* = \boldsymbol{\theta}_{0j}^* - 0.4I\left(\{j \in H_k\} \cap \{j \neq 1\}\right)$, $\boldsymbol{\theta}_{k1}^* = -0.4$; (b) dense contrasts: for each $k \in [5]$, let H_k be a random subset of $[p]$ with $|H_k| = p/2$ and let $\boldsymbol{\theta}_{kj}^* = \boldsymbol{\theta}_{0j}^* + \xi_j I(j \in H_k)$, where ξ_j is *i.i.d.* drawn from Laplace$(0, 0.04)$, $\boldsymbol{\theta}_{k1}^* = -0.4$.

Table 1: The means (standard deviations) of the simulation results for different methods under the settings in this section. For the column Datasets, by "Target" we mean only the target dataset is used, "Oracle" means we only use the useful datasets, and "All" means we use all the source datasets. In the case of using all datasets, if the method has dataset selection capability, we report the (TPR,TNR) of dataset selection instead of "All".

Estimator	Setting	$\| \cdot - \boldsymbol{\theta}_0^* \|_2$	$\| \cdot^\star - \boldsymbol{\theta}_0^* \|_2$	Datasets	Setting	$\| \cdot - \boldsymbol{\theta}_0^* \|_2$	$\| \cdot^\star - \boldsymbol{\theta}_0^* \|_2$	Datasets
$\widehat{\boldsymbol{\theta}}_{\text{target}}$	Ho_s	0.738 (0.103)	NA	Target	Ho_d	0.747 (0.095)	NA	Target
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$		0.737 (0.029)	0.374 (0.041)	Oracle		0.756 (0.036)	0.415 (0.048)	Oracle
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		1.088 (0.078)	0.838 (0.085)	All		1.145 (0.084)	0.901 (0.088)	All
$\widehat{\boldsymbol{\theta}}_{\text{Agg}}$		0.717 (0.078)	0.717 (0.078)	All		0.737 (0.095)	0.737 (0.095)	All
$\widehat{\boldsymbol{\theta}}_{\text{CV}}$		0.737 (0.029)	0.375 (0.043)	(1.00,1.00)		0.756 (0.036)	0.414 (0.047)	(1.00,1.00)
$\widehat{\boldsymbol{\theta}}_{\text{TF}}$		1.138 (0.013)	0.711 (0.068)	Oracle		1.152 (0.016)	0.723 (0.082)	Oracle
$\widehat{\boldsymbol{\theta}}_{\text{TN}}$		**0.381 (0.053)**	**0.324 (0.047)**	(1.00,1.00)		**0.451 (0.078)**	**0.399 (0.080)**	(0.99,1.00)
$\widehat{\boldsymbol{\theta}}_{\text{target}}$	He_s	0.787 (0.101)	NA	Target	He_d	0.788 (0.112)	NA	Target
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$		0.750 (0.032)	0.411 (0.057)	Oracle		0.773 (0.032)	0.449 (0.049)	Oracle
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		1.195 (0.091)	0.974 (0.104)	All		1.260 (0.089)	1.048 (0.095)	All
$\widehat{\boldsymbol{\theta}}_{\text{Agg}}$		0.793 (0.103)	0.793 (0.103)	All		0.790 (0.115)	0.790 (0.115)	All
$\widehat{\boldsymbol{\theta}}_{\text{CV}}$		0.750 (0.032)	0.413 (0.059)	(1.00,1.00)		0.773 (0.032)	0.451 (0.048)	(1.00,1.00)
$\widehat{\boldsymbol{\theta}}_{\text{TF}}$		1.141 (0.012)	0.761 (0.080)	Oracle		1.160 (0.017)	0.765 (0.086)	Oracle
$\widehat{\boldsymbol{\theta}}_{\text{TN}}$		**0.398 (0.102)**	**0.350 (0.105)**	(0.99,1.00)		**0.460 (0.050)**	**0.411 (0.044)**	(1.00,1.00)

In summary, for the informative sources, we have the following four settings: homogeneous covariates and sparse contrast vectors (with a small ℓ_1 norm), abbreviated as Ho_s; homogeneous covariates and dense contrast vectors (with a small ℓ_2 norm), denoted as Ho_d; heterogeneous covariates and sparse contrast vectors, denoted as He_s; and heterogeneous covariates and dense contrast vectors, denoted as He_d.

As for the non-informative datasets $k \in \mathcal{A}^c = \{6, \cdots, 10\}$, we consider: (a) larger sparse contrasts: for each $k \in \mathcal{A}^c$, let H_k be a random subset of $[p]$ with $|H_k| = 2s$, and let $\boldsymbol{\theta}^*_{kj} = \boldsymbol{\theta}^*_{0j} - 0.6I\left(\{j \in H_k\} \cap \{j \neq 1\}\right)$, $\boldsymbol{\theta}^*_{k1} = -0.4$; (b) larger dense contrasts: for each $k \in \mathcal{A}^c$, let H_k be a random subset of $[p]$ with $|H_k| = p/2$ and $\boldsymbol{\theta}^*_{kj} = \boldsymbol{\theta}^*_{0j} + \xi_j I(j \in H_k)$, where ξ_j is *i.i.d.* from Laplace$(0, 0.2)$, $\boldsymbol{\theta}^*_{k1} = -0.4$. The reason we set $\boldsymbol{\theta}^*_{k1} = -0.4$ for all $k \in [K]$ is to impose a systematic model shift in the first entry to highlight the effect of the fine-tuning step (Li et al., 2022).

We compare the following estimators and their fine-tuned versions, denoted by the additional superscript $\cdot^\star$. To avoid confusion, the proposed estimator obtained by solving (2.5) is denoted as $\widehat{\boldsymbol{\theta}}_{\mathrm{TN}}$ instead of $\widehat{\boldsymbol{\theta}}_0$, where TN stands for truncated norm. The competitors include the lasso estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{target}}$ using the target dataset; the oracle pooling estimator $\widehat{\boldsymbol{\theta}}_{\mathcal{P}}$ by pooling informative datasets $\mathcal{P}$; the TransFusion estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{TF}}$ by He et al. (2024); the blind pooling estimator $\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$ by pooling all datasets $\mathcal{P} \cup \mathcal{A}^c$; the aggregation estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{agg}}$ by Li et al. (2024); and the CV-based estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{CV}}$ by Tian and Feng (2023). We leave the implementation details of these competitors to the supplementary material.

We report the performances of these estimators in Table 1 based on 100 replications, where the truncated-penalized algorithm performs quite satisfactorily in its simultaneous estimation of the target parameter vector while identifying the useful sources. In addition, the truncated-penalized estimator also outperforms the oracle pooling estimator in some cases; this superiority is due to its capacity to estimate the target parameter and the contrasts simultaneously, as discussed in (Gross and Tibshirani, 2016; Ollier and Viallon, 2017; Li et al., 2024). Rigorous theoretical justification for such a phenomenon is beyond the scope of this work and is left for future pursuits.

5. Real Data Examples

In this section, we show the empirical usefulness of the proposed method in some real applications. We work on the following two cases concerning the IMDb movie reviews and the air quality in Beijing.

5.1 IMDb movie reviews

We first test our algorithm on a publicly available data set of movie reviews from `IMDb.com`, which is pre-processed and then used in Gross and Tibshirani (2016). The dataset contains 50,000 reviews of movies that have been split into training and testing datasets of the same size.

For each review, there is an integer rating ranging from 1 to 10, where 10 is the best. The dataset only contains positive reviews with a rating ≥ 7 and negative reviews with a rating ≤ 4. Following the procedures in Gross and Tibshirani (2016), we first use a binary bag of words representation of the reviews, using only words that were present in at least 500 reviews from the training set, resulting in $p = 993$ features.

We focus on the following seven genres of movies. The first three are the most commonly reviewed genres also used in Gross and Tibshirani (2016), namely Drama, Comedy, and Horror, with relatively large sample sizes of 4614, 2839, and 1441, respectively. We consider these three genres as sources. Then, we choose four genres, namely Action, Thriller, Sci-Fi, and Romance, with relatively small sample sizes of 1002, 655, 223, and 193, respectively, as potential targets. In each experiment, we take one of the latter four genres, say Action, as the target and treat the other six genres as potential sources of information.

We compare the same methods as in Section 4 with the same implementation details, and we report the mean squared prediction error on the test dataset. We report both $\mathrm{MSE}_{\mathrm{test}}$ and $\mathrm{MSE}^{\star}_{\mathrm{test}}$, where the subscript $\cdot^{\star}$ indicates that we use the fine-tuned estimator for the test dataset predic-

Table 2: Mean squared test dataset prediction error $\mathrm{MSE}_{\mathrm{test}}$ and $\mathrm{MSE}^{\star}_{\mathrm{test}}$, where the subscript $\cdot^{\star}$ indicates the fine-tuned version. We take the three most reviewed genres of movies, namely Drama, Comedy and Horror as sources (with relatively large sample sizes of 4614, 2839 and 1441, respectively). Then, we choose four genres, namely Action, Thriller, Sci-Fi and Romance (with relatively small sample sizes of 1002, 655, 223 and 193, respectively), as potential targets. In each experiment, we take one of the latter four genres, say Action, as the target, and treat the other six genres as potential sources.

Estimator	Target	$\mathrm{MSE}_{\mathrm{test}}$	$\mathrm{MSE}^{\star}_{\mathrm{test}}$	Estimator	Target	$\mathrm{MSE}_{\mathrm{test}}$	$\mathrm{MSE}^{\star}_{\mathrm{test}}$
$\widehat{\boldsymbol{\theta}}_{\mathrm{target}}$		7.949	NA	$\widehat{\boldsymbol{\theta}}_{\mathrm{target}}$		11.236	NA
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		8.032	8.032	$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		8.347	8.858
$\widehat{\boldsymbol{\theta}}_{\mathrm{CV}}$	Action	8.032	8.026	$\widehat{\boldsymbol{\theta}}_{\mathrm{CV}}$	Thriller	8.353	8.844
$\widehat{\boldsymbol{\theta}}_{\mathrm{TF}}$		8.020	7.747	$\widehat{\boldsymbol{\theta}}_{\mathrm{TF}}$		8.316	8.290
$\widehat{\boldsymbol{\theta}}_{\mathrm{TN}}$		**7.423**	**7.423**	$\widehat{\boldsymbol{\theta}}_{\mathrm{TN}}$		**7.386**	**8.061**
$\widehat{\boldsymbol{\theta}}_{\mathrm{target}}$		11.433	NA	$\widehat{\boldsymbol{\theta}}_{\mathrm{target}}$		8.369	NA
$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		7.778	7.778	$\widehat{\boldsymbol{\theta}}_{\mathcal{P}\cup\mathcal{A}^c}$		5.672	6.396
$\widehat{\boldsymbol{\theta}}_{\mathrm{CV}}$	Sci-Fi	7.778	7.778	$\widehat{\boldsymbol{\theta}}_{\mathrm{CV}}$	Romance	5.657	6.365
$\widehat{\boldsymbol{\theta}}_{\mathrm{TF}}$		7.454	7.454	$\widehat{\boldsymbol{\theta}}_{\mathrm{TF}}$		**5.262**	6.248
$\widehat{\boldsymbol{\theta}}_{\mathrm{TN}}$		**7.085**	**7.085**	$\widehat{\boldsymbol{\theta}}_{\mathrm{TN}}$		5.750	**6.089**

tion. We do not report the aggregation estimator from Li et al. (2024) here due to its inherent randomness and instability arising from sample splitting in a single experiment. Its performance is, after all, not competitive in this particular data case. The results are similar to those in Section 4, where our method has satisfactory performance across various settings. In the real data case, the truncated-penalized estimator outperforms the other estimators on the test set in three out of four cases, except when Romance is the target and the TransFusion estimator has the smallest test dataset mean squared error.

5.2 Air pollution data

Air pollution is an urgent global environmental issue that attracts significant attention from countries worldwide. The majority of the problem is caused by human activities, such as industrial emissions and vehicular exhaust. These activities release various pollutants, such as particulate matter (PM), nitrogen oxides (NO_x), sulfur dioxide (SO_2), and greenhouse gasses into the air. The detrimental effects of air pollution are wide-ranging and severe, posing significant health risks, contributing to environmental degradation, climate change, and the deterioration of ecosystems.

From the perspective of temporal dependence, air pollution may easily undergo dramatic changes due to specific events, such as changes in weather and human interventions. That is, there exist frequent change points as time goes by, which result in relatively short time windows for prediction tasks. Besides, if we apply high frequency data to increase the sample size, the excessive dependencies may undermine the effectiveness of our models. Hence, we collect the daily datasets as a trade-off between dependencies and sample size. From the perspective of spatial dependence, a notable feature is geographical similarity; that is, geographically adjacent regions may exhibit similar air quality as a result of the diffusion of air pollution. This feature encourages us to employ information from adjacent regions to increase the sample size, in the same spirit of transfer learning.

We use transfer learning to analyze the air pollution dataset in Beijing, China. We aim to enhance the next-day prediction performance and provide insights into the winter air pollution problem in Beijing using the proposed method. We collect datasets from the targeted Beijing site and eight potentially useful cities: Tianjin (TJ), Shijiazhuang (SJZ), Tangshan (TS), Zhangjiakou (ZJK), Hefei (HF), Nanchang (NC), Wuhan (WH), and Shenzhen (SZ). Note that the first four cities are geographically adjacent to Beijing, which might intuitively suffer from similar patterns of air pollution as Beijing.

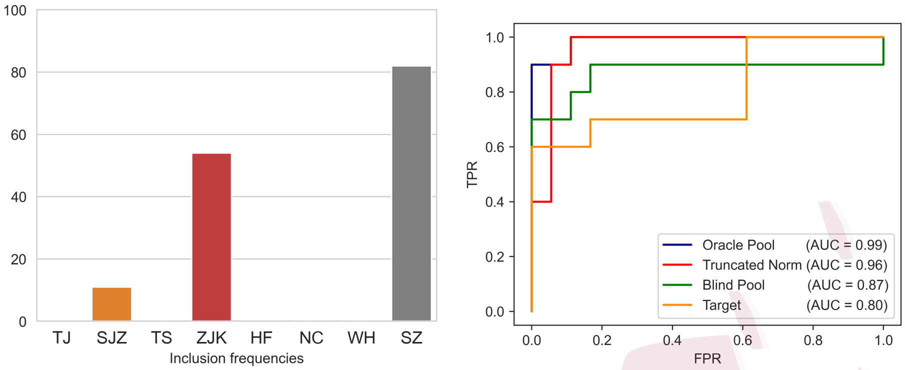

Figure 1: Inclusion frequencies of each source dataset in estimating the daily parameters of the Beijing site using the truncated-penalized algorithm (left). The two most relevant sites (ZJK and SZ) are then used as the oracle informative source datasets in the backtracking rolling windows. The prediction accuracy by various methods are then reported using the receiver operating characteristic (ROC) curve and the area under curve (AUC) metric (right).

For each target and source dataset, we collect the daily data from January to February 2019. On each day t, the covariates $\boldsymbol{X}_{k,t}$ are matrix-valued data where the rows represent 24 hours and the columns represent the content of six common air pollutants: $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, and CO. The response $y_{k,t}$ is a binary variable, where 1 represents mild pollution and 0 represents relatively good air quality. Clearly, the matrix-valued covariates possess certain column-wise and row-wise correlations, and we add a nuclear norm penalty to obtain the low-rank estimation. Specifically, we model the next-day air quality by

$$\mathbb{P}(y_{k,t+1}|\boldsymbol{X}_{k,t}) \propto \exp\left\{y_{k,t+1}\langle\boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,t}\rangle - b(\langle\boldsymbol{\theta}_k^*, \boldsymbol{X}_{k,t}\rangle)\right\},$$

for the logit link $b'(x) = 1/(1+e^{-x})$, using the rolling windows approach with a window size of 31 to forecast air pollution for February (28 days). For each day, we set $K = 8$, $n_k = n_0 = 31$, $p_1 = 24$, $p_2 = 6$, and respectively use the vanilla target estimator, the blind pooling estimator, and the truncated penalized estimator for constructing the next-day prediction.

We report the inclusion frequencies of each source dataset in estimating the daily parameters of the Beijing site using the truncated-penalized algorithm, as shown in the left panel of Figure 1. This suggests that both Zhangjiakou (ZJK), geographically adjacent to Beijing, and Shenzhen (SZ) are informative auxiliary datasets for prediction. Here, Shenzhen might have been chosen due to similarities in industrial structure, population, and other social factors with Beijing, which implies that selecting auxiliary datasets based on geographical proximity alone may not be sufficient for the air pollution prediction problem. Then, these two sites (ZJK and SZ) are used as the oracle informative source datasets in the backtracking rolling windows. Note that the information about the useful datasets is not obtained until the end of the month. We could see that the oracle pooling estimator from backtracking achieves the highest area under the curve (AUC) score (AUC=0.99) on the right panel of Figure 1, while the truncated-penalized estimator performs comparably (AUC=0.96).

6. Discussion

In this work, we propose estimating the target parameter vector while simultaneously selecting the informative sources with a non-convex penalty. The proposed algorithm is justified from both statistical and computational aspects. For future work, it is interesting but also challenging to investigate the theoretical properties of the methods that simultaneously estimate the target parameter and the contrasts; see, for example, (Gross and Tibshirani, 2016; Ollier and Viallon, 2017; Li et al., 2024).

Supplementary Material

In the supplementary material, we first provide extensional theoretical arguments concerning the two specific statistical models and remark on the optional fine-tuning step. Then, we present additional numerical details that further support our arguments. Finally, we provide the proofs of the theoretical results.

References

Bastani, H. (2021). Predicting with proxies: Transfer learning in high dimension. *Management Science 67*(5), 2964–2984.

Boyd, S., N. Parikh, and E. Chu (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Now Publishers Inc.

Cai, J.-F., E. J. Candès, and Z. Shen (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization 20*(4), 1956–1982.

Cai, T. T. and H. Wei (2021). Transfer learning for nonparametric classification. *The Annals of Statistics 49*(1), 100–128.

Candes, E. and T. Tao (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313–2351.

Chen, S., Q. Zheng, Q. Long, and W. J. Su (2021). A theorem of the alternative for personalized federated learning. *arXiv preprint arXiv:2103.01901*.

Duan, Y. and K. Wang (2023). Adaptive and robust multi-task learning. *The Annals of Statistics 51*(5), 2015–2039.

Fan, J., W. Gong, and Z. Zhu (2019). Generalized high-dimensional trace regression via nuclear norm regularization. *Journal of Econometrics 212*(1), 177–202.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association 96*(456), 1348–1360.

Fan, J., W. Wang, and Z. Zhu (2021). A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery. *The Annals of Statistics 49*(3), 1239 – 1266.

Gross, S. M. and R. Tibshirani (2016). Data shared lasso: A novel tool to discover uplift. *Computational statistics & data analysis 101*, 226–235.

Hamidi, N. and M. Bayati (2022). On low-rank trace regression under general sampling distribution. *Journal of Machine Learning Research 23*(321), 1–49.

He, Y., X. Kong, L. Trapani, and L. Yu (2023). One-way or two-way factor model for matrix sequences? *Journal of Econometrics 235*(2), 1981–2004.

He, Z., Y. Sun, and R. Li (2024). Transfusion: Covariate-shift robust transfer learning for high-dimensional regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 703–711. PMLR.

Li, S., T. T. Cai, and H. Li (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology 84*(1), 149–173.

Li, S., L. Zhang, T. T. Cai, and H. Li (2024). Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association 119*(546), 1274–1285.

Liu, D., C. Zhao, Y. He, L. Liu, Y. Guo, and X. Zhang (2023). Simultaneous cluster structure learning and estimation of heterogeneous graphs for matrix-variate fmri data. *Biometrics 79*(3), 2246–2259.

McMahan, B., E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR.

Negahban, S. N., P. Ravikumar, M. J. Wainwright, and B. Yu (2012). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science 27*(4), 538–557.

Niu, S., Y. Liu, J. Wang, and H. Song (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence 1*(2), 151–166.

Ollier, E. and V. Viallon (2017). Regression modelling on stratified data with the lasso. *Biometrika 104*(1), 83–96.

Pan, W., X. Shen, and B. Liu (2013). Cluster analysis: Unsupervised learning via supervised learning with a non-convex penalty. *Journal of Machine Learning Research 14*(7), 1865–1889.

Shen, X., W. Pan, and Y. Zhu (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association 107*(497), 223–232.

Sun, Y., P. Babu, and D. P. Palomar (2016). Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing 65*(3), 794–816.

Thi Hoai An, L. and P. Dinh Tao (1997). Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of global optimization 11*(3), 253–285.

Tian, Y. and Y. Feng (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association 118*(544), 2684–2697.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology 58*(1), 267–288.

Torrey, L. and J. Shavlik (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264. IGI global.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, Volume 3. Cambridge university press.

Wu, C., S. Kwon, X. Shen, and W. Pan (2016). A new algorithm and theory for penalized regression-based clustering. *Journal of Machine Learning Research 17*(188), 1–25.

Zhou, H. and L. Li (2014). Regularized matrix regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology 76*(2), 463–483.

Zhuang, F., Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE 109*(1), 43–76.