Statistical Multi-Task Learning for Robust Task Clustering via Fused Regularization

岡﨑, 彰良

Statistical Multi-Task Learning for Robust Task Clustering via Fused Regularization

Akira Okazaki

Graduate School of Mathematics

Kyushu University

2025

Abstract

Multi-task learning (MTL) is a methodology where we simultaneously estimate multiple related tasks to improve their estimation and prediction accuracy. In the light of statistical modeling, MTL methods are formulated as a joint optimization problem with the sum of the loss functions coming from multiple datasets and regularization terms facilitating the transfer of common information among tasks. In this dissertation, we consider regression problems as estimation tasks and the situation where the observed task set consists of tasks with heterogeneous characteristics. MTL methods treating this situation are based on clustering, which groups the task set into some homogeneous clusters by their latent common characteristics. This enables us to improve estimation accuracy by transferring common information within each cluster. However, existing clustering-based MTL methods typically rely on the fused regularization term, which still causes the incorrect transfer of information among tasks with different characteristics.

To overcome this problem, we propose two novel multi-task learning methods. First, we propose an MTL method using regularization terms based on convex clustering. This method separates the parameters into those for regression models and task clustering. While the regularization term fuses the parameters for task clustering, the regression parameters are not directly fused. Therefore, we can expect to reduce the incorrect transfer of information for the regression parameters. Second, we propose an MTL method that simultaneously estimates cluster structure and detects outlier tasks with unique characteristics. To reduce the contamination in the cluster estimation caused by outlier tasks, we introduce parameters representing outlier components within the task. These parameters are selected from regularization terms inducing group sparsity. Furthermore, we also construct the relationship between the formulation given by outlier parameter selection and the M-estimator in the context of robust statistics. This gives an interpretation of robustness towards outlier tasks. The effectiveness of the proposed methods is demonstrated through numerical simulations and application to the real dataset.

要約

マルチタスク学習とは，複数の推定タスクが存在する場合において，それらの統合に基づき個々のタスクの推定精度を改善する方法論である．統計的モデリングの観点においては，マルチタスク学習は複数のタスクに由来する損失関数の和に対して，タスク間での情報の共有を誘引する正則化項および制約条件を課した最適化問題として定式化される．本論文では，推定タスクに回帰問題を想定し，得られているタスク集合に異質な特徴を持つタスクが混在している状況に着目する．このような状況においては，クラスタリングに基づきタスク集合を同一の特徴を有するクラスタへ分類することにより，推定精度を改善することが可能である．具体的には，連結型の正則化項によって類似するタスクのパラメータを統合することにより，タスクに対するクラスタリングを実行する．しかし，既存の連結型の正則化に基づく手法では，類似性が低いタスク間に対してもパラメータが統合される働きが生じ，推定精度の悪化を招く．

本論文では，異なる特徴を有するタスク間における統合を軽減するため，二つの新たな正則化法に基づくマルチタスク学習手法を提案する．まず一つ目は，凸クラスタリングに基づくマルチタスク学習手法である．この手法では，個々のタスクに関しての回帰係数に加え，タスクのクラスタリングに用いる重心パラメータを導入する．重心パラメータは連結型の正則化によりタスク間で統合される一方，各タスクの回帰係数は他タスクの回帰係数と直接統合されない．これにより，各タスクの回帰係数が異なる特徴を有するクラスタから受ける影響を軽減することが可能となる．二つ目は，タスクに関するクラスタ構造の推定と，独自性が大きい外れ値タスクの検出を同時に行う手法である．この手法では，外れ値タスクがクラスタ構造の推定に及ぼす影響を軽減するため，各タスクに対して外れ値パラメータを導入する．外れ値タスクは，この外れ値パラメータを内包するか否かで判断することができ，パラメータの選択はグループ正則化法により実行される．さらに，本手法と，ロバスト推定における M-推定量に基づく方法との関係についても述べる．提案した手法に対して，モンテカルロ・シミュレーションと実データへの適用を通じて有効性を示す．

Acknowledgements

This dissertation was completed not only by my own work but also with the invaluable support and guidance of many people. First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Shuichi Kawano, for his excellent guidance, insightful feedback, and continuous encouragement throughout my doctoral studies. His dedication and thoughtful mentorship have profoundly shaped my development as a researcher. I am also deeply grateful to Professor Kei Hirose, Professor Koji Tsukuda, and Professor Hirokazu Yanagihara for their valuable advice, constructive comments, and suggestions.

My sincere appreciation goes to all members of the Kawano, Hirose, and Tsukuda laboratories. Their stimulating discussions, collaborative environment, and friendly atmosphere have been essential to my research journey. Their support and camaraderie made my time as a doctoral student productive and enjoyable.

This research was supported by JST SPRING, Grant Number JPMJSP2136, for which I am very thankful. The computational resource was provided by the Super Computer System, Human Genome Center, Institute of Medical Science, The University of Tokyo.

Finally, I would like to express my heartfelt gratitude to my mother and family for their support, understanding, and encouragement throughout my academic pursuits.

Akira Okazaki

February, 2025, Fukuoka

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Recent improvements in computational capabilities and increased attention to statistics and machine learning have led to rapid growth in large-scale databases, such as the Federal Reserve Economic Data (FRED) in economics and the National Center for Biotechnology Information (NCBI) in life sciences. Machine learning approaches, particularly deep learning, have successfully leveraged those extensive datasets for remarkable prediction accuracy. However, similar advancements in statistical methodology have been comparatively limited. Developing more sophisticated statistical frameworks utilizing large-scale data would enable us to extract reliable and interpretable information efficiently.

Multi-task learning (MTL) (Caruana, 1997) is a general framework of statistics and machine learning where we simultaneously learn multiple related tasks so that each task leverages information from other tasks. Because, in real problems, related tasks tend to have the same common information, MTL can lead to better performance than independently estimating each task (Zhang and Yang, 2021). Because of this advantage, MTL has been applied to many problems in various fields of research, such as disease progression prediction (Zhou et al., 2011b), biomedicine (Li et al., 2018), transportation (Deng et al., 2017), image annotation (Fan et al., 2008), speech recognition

(Parameswaran and Weinberger, 2010), and so on.

From a statistical modeling perspective, MTL methods are formulated as an optimization problem that integrates multiple statistical models corresponding to multiple datasets and prior information about relationships among models. This information is incorporated as a regularization term or constraint for model parameters to transfer the information among tasks. Thus, the effectiveness of MTL methods critically depends on how these task relationships are modeled. When tasks exhibit the assumed relationships, MTL can leverage shared information to improve estimation efficiency and generalization performance. In general, MTL methods are roughly classified into two main categories according to the assumption of task relationships. The first is to assume that all tasks share a common structure. This approach is achieved by estimating low-rank representation (Ando and Zhang, 2005), sparsity pattern (Obozinski et al., 2010), and so on. For example, deep learning models with MTL are often formulated as sharing the same hidden parameters among tasks and learning them. In some practical situations, it is difficult to assume that all tasks have the same structure. When tasks are unrelated or weakly related, forcing information sharing among them can be detrimental. The second category addresses this issue by assuming that tasks can be classified into some latent groups sharing common characteristics (Kang et al., 2011). MTL methods based on this assumption are achieved by clustering the parameters of models. For instance, Argyriou et al. (2007) introduced the k-means algorithm for task clustering, while several other studies (Zhong and Kwok (2012); Yamada et al. (2017); He et al. (2019); Dondelinger et al. (2020)) utilized fused regularization techniques such as fused lasso (Tibshirani et al., 2005) and network lasso (Hallac et al., 2015). However, existing methods based on the clustering approach often suffer from negative transfer between irrelevant tasks, which worsens the estimation and prediction accuracy of each task.

In this thesis, we address two challenges in clustering-based MTL methods. First,

we examine the limitations of fused regularization approaches to task clustering. While these methods offer the advantage of convex optimization with guaranteed global optima, their regularization terms can force undesirable shrinkage between parameters of tasks that should belong to different clusters, leading to incorrect information transfer. To address this, we propose Multi-Task Learning via ConVeX clustering (MTLCVX), which introduces a novel regularization based on convex clustering. MTLCVX shrinks centroid parameters representing cluster centers while estimating model parameters around these centroids. This reduces negative transfer between tasks while maintaining the benefits of convex optimization.

Second, we tackle the challenge of outlier tasks, which are either highly unique or share no common characteristics with others. Traditional clustering techniques in MTL attempt to assign all tasks to clusters, which can lead to deteriorated clustering performance and misspecified relationships when outlier tasks are present. Although, some robust MTL methods (Chen et al., 2011; Gong et al., 2012) have been proposed to handle these outlier tasks, they typically impose tasks to have a single shared structure and outlier components. Furthermore, they often employed group lasso regularization (Yuan and Lin, 2006) that may overly constrain outlier parameters. To overcome these limitations, we propose Multi-Task Learning with Robust Regularized Clustering (MTLRRC), which simultaneously performs task clustering and outlier detection through robust regularization terms based on robust convex clustering (Quan and Chen, 2020). MTLRRC extends this framework to incorporate non-convex and group-sparse penalties, enabling effective outlier identification. We establish connections between our approach and multivariate M-estimators, which provide an intuitive interpretation of the robustness of MTLRRC against outlier tasks. The method is implemented through a modified alternating direction method of multipliers (ADMM; (Boyd et al., 2011)). Those comprehensive theoretical and empirical convergence analyses are also provided.

The remainder of this thesis is structured as follows.

- **Chapter 2** describe the existing multi-task learning methods. First, we explain the general problem setup of MTL. Then, we introduce the main approaches and their specific methods. Moreover, some relationships within them are also given.
- **Chapter 3** describes the convex clustering and the estimation algorithm.
- **Chapter 4** discusses the problem caused by the fused regularization term, and the multi-task learning method via convex clustering is proposed. The effectiveness of the proposed method is shown through Monte Carlo simulations and applications to real data.
- **Chapter 5** discusses the clustering of tasks with outlier tasks, and a robust MTL method is proposed. The non-convex extension of the robust convex clustering and the connection to the multivariate M-estimator is also given. The effectiveness of the method is demonstrated through simulation studies and application to real data.
- **Chapter 6** provides concluding remarks and discusses future research directions.

Notations

In this thesis, we use capital letters to represent matrices. For a matrix $Z \in \mathbb{R}^{a\times b}$, we denote the row vectors by boldface small letters with subscripts $\boldsymbol{z}_i$ and the column vectors by boldface small letters with superscripts $\boldsymbol{z}^j$, such that $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_a)^\top = (\boldsymbol{z}^1, \ldots, \boldsymbol{z}^b)$. For a vector $\boldsymbol{z} = (z_1, \ldots, z_a) \in \mathbb{R}^a$, we define the L_q^l-norm as $\|\boldsymbol{z}\|_q^l = (\sum_{i=1}^a |z_i|^q)^{\frac{l}{q}}$. When $l = 1$, we may omit the superscript for simplicity. For a matrix $Z \in \mathbb{R}^{a\times b}$, we define:

- The row-wise L_q^l-norm (denoted as $L_{1,q}^l$-norm): $\|Z\|_{1,q}^l = \sum_{i=1}^a \|\boldsymbol{z}_i\|_q^l$.
- The column-wise L_q^l-norm (denoted as $L_{2,q}^l$-norm): $\|Z\|_{2,q}^l = \sum_{j=1}^b \|\boldsymbol{z}^j\|_q^l$.

- The trace operator: $\mathrm{tr}(Z) = \sum_{i=1}^{a} Z_{ii}, \quad (b = a)$.
- The Frobenius norm: $\|Z\|_F = \mathrm{tr}(Z^\top Z)^{1/2}$.

We denote the maximum eigenvalue by $\lambda_+(Z)$ and the strictly positive minimum eigenvalue by $\lambda_{++}(Z)$. Additionally, we define the vectorization operator as $\mathrm{vec}(Z) = (\boldsymbol{z}_1^\top, \boldsymbol{z}_2^\top, \ldots, \boldsymbol{z}_a^\top)^\top \in \mathbb{R}^{ab}$, and inner product as $\langle \cdot, \cdot \rangle$ that represents $\boldsymbol{a}^\top \boldsymbol{b}$ for vectors and $\mathrm{tr}(A^\top B)$ for matrices.

We denote "subject to" as "s.t." for the abbreviation in equations and minimization problems. "Eq." and "Eqs." are used for abbreviations of "equation" and "equations", respectively.

Chapter 2

Multi-task learning with generalized linear models

Multi-task learning is a general framework that includes a wide range of methods, which aim to learn or estimate group-specific parameters corresponding to multiple sets of samples. For example, in statistics, multivariate regression and multi-class logistic regression can be viewed as special cases of multi-task learning, because they have parameter vectors for each pair of a feature's response vector and a design matrix. In this chapter, we first describe the general problem setup formulated as multiple generalized linear models (GLMs), which we address through this thesis. Then, we introduce several multi-task learning methods by classifying them into some approaches.

2.1 General problem setup

Suppose that we have T datasets. For each dataset m $(m = 1, \ldots, T)$, we observed n_m pairs of data points $\{(\boldsymbol{x}_{mi}, y_{mi}); i = 1, \ldots, n_m\}$, where $\boldsymbol{x}_{mi}$ is a p-dimensional explanatory variables and y_{mi} is the corresponding response variable following distribution in the exponential family with mean $\mu_{mi} = \mathbb{E}[y_{mi}|\boldsymbol{x}_{mi}]$. Let $X_m = (\boldsymbol{x}_{m1}, \ldots, \boldsymbol{x}_{mn_m})^\top \in \mathbb{R}^{n_m \times p}$ be the design matrix, $\boldsymbol{y}_m = (y_{m1}, \ldots, y_{mn_m})^\top \in \mathbb{R}^{n_m}$ be the response vector

for dataset m, and $n = \sum_{m=1}^{T} n_m$ be the total number of samples. We assume that each feature vector $\boldsymbol{x}_{mi}$ is standardized to have zero mean and unit variance, which is essential to compare the model parameters among tasks. The centered response vector $\boldsymbol{y}_m$ with zero mean is also assumed, when it is continuous.

Our goal is to estimate T generalized linear models (GLMs) simultaneously, which take the form:

$$\eta_{mi} = g(\mu_{mi}) = w_{m0} + \boldsymbol{x}_{mi}^{\top} \boldsymbol{w}_m, \quad i = 1, \ldots, n_m, \ m = 1, \ldots, T, \tag{2.1}$$

where w_{m0} is an intercept for m-th task, $\boldsymbol{w}_m = (w_{m1}, \ldots, w_{mp})^{\top}$ is a p-dimensional regression coefficient vector for m-th task, η_{mi} is a linear predictor, and $g(\cdot)$ is a canonical link function. We assume that all task's response variables given by the explanatory variables follow the same type of distribution expressed as

$$f(y_{mi}|\boldsymbol{x}_{mi}; \theta(\boldsymbol{x}_{mi}), \phi) = \exp\left\{ \frac{y_{mi}\theta(\boldsymbol{x}_{mi}) - b(\theta(\boldsymbol{x}_{mi}))}{a(\phi)} + c(y_{mi}, \phi) \right\},$$

where $a(\cdot), b(\cdot)$, and $c(\cdot)$ are known functions that vary according to the distributions, ϕ is a known dispersion parameter, and $\theta(\cdot)$ is the natural parameter, which is expressed as $g(\mu_{mi}) = \theta(\boldsymbol{x}_{mi})$.

Let $W = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T)^{\top} \in \mathbb{R}^{T \times p}$ be the regression coefficient matrix, and $\boldsymbol{w}_0 = (w_{10}, \ldots, w_{T0})^{\top} \in \mathbb{R}^T$ be the intercept vector. To estimate T GLMs in (2.1) simultaneously, we formulate multi-task learning (MTL) methods as

$$\min_{\boldsymbol{w}_0, W} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \Omega(W) \right\}, \tag{2.2}$$
$$\text{s.t. } W \in \mathcal{W},$$

where $L(\cdot, \cdot)$ is a loss function derived from the negative log-likelihood of a GLM task, $\Omega(\cdot)$ is a regularization term that encourages sharing the information among tasks, and $\mathcal{W}$ is a parameter space representing constraints that facilitate underlying task

structure. When $\Omega(W) = 0$ and $\mathcal{W} = \mathbb{R}^{T \times p}$, Problem (2.2) is the same as independently estimating ordinary GLMs.

When continuous response vectors $\boldsymbol{y}_m \in \mathbb{R}^{n_m}$ are considered, the linear regression is given by $a(\phi) = \phi, \phi = 1$, and $b(\theta) = \frac{\theta^2}{2}$. The regression loss function is

$$L(w_{m0}, \boldsymbol{w}_m) = \frac{1}{2}\|\boldsymbol{y}_m - \boldsymbol{X}_m \boldsymbol{w}_m\|_2^2. \tag{2.3}$$

Note that the intercepts are excluded from the model without loss of generality. If only a design matrix $X = X_1 = X_2 = \cdots = X_T \in \mathbb{R}^{n_0 \times p}$ and its multiple response vectors $\{\boldsymbol{y}_m \in \mathbb{R}^{n_0}; m = 1, \ldots, T\}$ are observed, Problem (2.2) becomes the optimization problem of the multivariate regression:

$$\min_W \left\{ \frac{1}{2n}\|Y - XW^\top\|_F^2 + \Omega(W) \right\},$$
$$\text{s.t. } W \in \mathcal{W},$$

where $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \in \mathbb{R}^{n_0 \times T}$. Therefore, MTL methods can be applied to the datasets usually analyzed by multivariate regression models. There is one different aspect between the MTL setting and multivariate regression. In multivariate regression, since a sample is observed as pairs of multivariate response and explanatory variables, the correlations in response features and residuals can be taken into account for the model estimation. For instance, the envelope models (e.g. Cook et al. (2010); Lee and Su (2020)) use this information to restrict the parameter space $\mathcal{W}$ and gain efficiency for the estimation of regression coefficients. On the other hand, samples differ between tasks in the MTL setting, and the correlation information of responses does not exist.

When binary response vectors $\boldsymbol{y}_m \in \{0, 1\}^{n_m}$ are considered, the logistic regression is given by $a(\phi) = \phi, \phi = 1$, and $b(\theta) = \log(1 + e^\theta)$. The loss function of logistic regression is

$$L(w_{m0}, \boldsymbol{w}_m) = -\sum_{i=1}^{n_m} \left\{ y_{mi}(w_{m0} + \boldsymbol{w}_m^\top \boldsymbol{x}_{mi}) - \log\{1 + \exp(w_{m0} + \boldsymbol{w}_m^\top \boldsymbol{x}_{mi})\} \right\}. \tag{2.4}$$

Similar to multivariate regression, MTL methods can also be applied to multi-class logistic regression. Suppose that categorical response g taking values in $\mathcal{G} = \{1, \ldots, T\}$ are observed. Multi-class logistic regression is modeled as

$$\Pr(g = k|\boldsymbol{x}) = \frac{\exp(w_{k0} + \boldsymbol{w}_k^\top \boldsymbol{x})}{\sum_{m=1}^{T} \exp(w_{m0} + \boldsymbol{w}_m^\top \boldsymbol{x})},$$

where w_{m0} and $\boldsymbol{w}_m$ are intercept and a regression coefficient vector for m-th class. Let Y be the $T \times n_0$ indicator response matrix with elements $y_{ki} = I(g_i = k)$. Then, the negative log-likelihood is given by

$$L(\boldsymbol{w}_0, W) = -\sum_{m=1}^{T}\sum_{i=1}^{n_0} \left\{ y_{mi}(w_{m0} + \boldsymbol{w}_m^\top \boldsymbol{x}_i) - \log\{1 + \exp\left(w_{m0} + \boldsymbol{w}_m^\top \boldsymbol{x}_i\right)\}\right\}. \tag{2.5}$$

By comparing Eqs. (2.4) and (2.5), the difference is only task indices of explanatory variables. Therefore, we can apply MTL methods to multi-class logistic regression, and the parameters are computed in the same way as multi-task binary logistic regression.

2.2 Low-rank approach

In the low-rank approach, the regression coefficient matrix W is assumed to have a low-rank structure. In other words, a task's regression coefficient vector is represented as a linear combination of the other task's regression coefficient vectors. By estimating the low-rank structure, we can reduce the total number of parameters and the model complexity.

Ando and Zhang (2005) proposed the following MTL method:

$$\begin{aligned} &\min_{\boldsymbol{w}_0, W, U, \Gamma} \sum_{m=1}^{T} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|\boldsymbol{v}_m\|_2^2 \right\}, \\ &\text{s.t. } \boldsymbol{w}_m = \Gamma^\top \boldsymbol{u}_m + \boldsymbol{v}_m, \quad m = 1, \ldots, T, \quad \Gamma\Gamma^\top = I_h, \end{aligned} \tag{2.6}$$

where Γ is an $h \times p$ shared matrix, $\boldsymbol{u}_m \in \mathbb{R}^h, \boldsymbol{v}_m \in \mathbb{R}^p$ are latent task parameters, λ is a regularization parameter with non-negative value, and h $(< p)$ is a prespecified

non-negative integer. From the constraint, the regression coefficient vector $\boldsymbol{w}_m$ is decomposed into the task relation part spanned by shared orthogonal basis Γ with the weight parameter $\boldsymbol{u}_m$ and task-specific part $\boldsymbol{v}_m$. By substituting $\boldsymbol{v}_m = \boldsymbol{w}_m - \Gamma^\top \boldsymbol{u}_m$, Problem (2.6) is reformulated as

$$\min_{\boldsymbol{w}_0, W, U, \Gamma} \sum_{m=1}^{T} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|\boldsymbol{w}_m - \Gamma^\top \boldsymbol{u}_m\|_2^2 \right\},$$
$$\text{s.t.} \quad \Gamma\Gamma^\top = I_h.$$

If we consider minimizing the problem concerning $\boldsymbol{u}_m$, the optimal $\widehat{\boldsymbol{u}}_m = \Gamma \boldsymbol{w}_m$ is given by the first-order condition. Thus, we can reformulate the problem as

$$\min_{\boldsymbol{w}_0, W, \Gamma} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda (\|W\|_F^2 - \operatorname{tr}(W\Gamma^\top \Gamma W^\top)) \right\}, \tag{2.7}$$
$$\text{s.t.} \quad \Gamma\Gamma^\top = I_h.$$

Then, this optimization problem can be solved by alternating optimization procedure, which is done by alternately optimizing Problem (2.7) concerning $(\boldsymbol{w}_0, W)$ with fixed Γ and Γ with fixed $(\boldsymbol{w}_0, W)$. In particular, the optimization problem concerning Γ is represented as

$$\max_{\Gamma} \operatorname{tr}(\Gamma W^\top W \Gamma^\top), \quad \text{s.t.} \quad \Gamma\Gamma^\top = I_h. \tag{2.8}$$

This problem is the same formulation as in the principal component analysis (PCA), where each task regression vector $\boldsymbol{w}_m$ is considered as a sample. Therefore, the optimal $\widehat{\Gamma}$ with fixed W is given by the h eigenvectors corresponding to the largest h eigenvalues of the task covariance matrix $W^\top W$. Consequently, the problem jointly estimates a low-rank orthogonal basis whose linear combination can predict all task responses by minimizing task-specific part $\boldsymbol{v}_m$.

Because Problem (2.6) contains a non-convex optimization problem (2.8) for each update of Γ, the alternative procedure is not guaranteed to find a global optima. To address this problem, many MTL and multivariate regression methods (e.g. Argyriou

et al. (2006); Pong et al. (2010)) adopted the following formulation:

$$\min_{\boldsymbol{w}_0, W} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|W\|_* \right\}. \tag{2.9}$$

Here, the second term is known as nuclear or trace norm regularization representing $\|W\|_* = \sum_{i=1}^{\min(T,p)} \sigma_i(W)$, where $\sigma_i(\cdot)$ denote the i-th singular value of a matrix. Because the nuclear norm regularization can be viewed as an L_1-norm regularization for singular values, some are estimated to have zero value, which also means the estimated $\widehat{W}^\top \widehat{W}$ become a low-rank matrix. Thus, Problem (2.9) is a convex relaxation of Ando and Zhang (2005).

2.3 Sparse approach

Variable selection is a fundamental approach in statistical modeling that extracts useful information from high-dimensional datasets. It is known that even in high dimensional situations where $n \ll p$, it is possible to select the true meaningful variables and predict the future response, if those variables are sufficiently sparse in the variables (e.g. Hastie et al. (2015)). In general, if the value of the regression coefficient is zero, the variable is considered meaningless, because the corresponding feature has no contribution to the response variable. To estimate redundant regression coefficients to exactly zero, the lasso (Tibshirani, 1996) is widely used in single-task learning settings, which take the form:

$$\min_{w_0, \boldsymbol{w}} \left\{ \frac{1}{n} L(w_0, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1 \right\}.$$

By imposing L_1-norm as the regularization term, the lasso shrinks the value of the regression coefficients towards zero. Consequently, we can obtain a sparse solution consisting of estimated meaningful variables. In the multi-task learning setting, multiple regression coefficient vectors having the same features are considered. Intuitively, those same variables may be useful between similar tasks. For example, consider an estimation task of a handwritten word by a writer with features of pixels. There would be

commonly used pixels and unused pixels among many writers, which may represent the characteristics of the letter itself. Those important features would be found by selecting a subset of commonly shared variables.

To select important variables with all tasks, Turlach et al. (2005) and Liu et al. (2009) considered the following sparse MTL method:

$$\min_{\boldsymbol{w}_0, W} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|W\|_{2,\infty} \right\}. \tag{2.10}$$

The second term is an $L_{2,\infty}$-norm regularization representing $\|W\|_{2,\infty} = \sum_{j=1}^{p} \max(|w_{1j}|, \ldots, |w_{Tj}|)$. The $L_{2,\infty}$ regularization groups j-th variable across all tasks, and each maximum value is regularized by the L_1-norm. If the regression coefficient with the largest value in the j-th variable is estimated to be zero, the same variables in other tasks also go to zero value, simultaneously.

For the same motivation, Lounici et al. (2009) and Obozinski et al. (2010) proposed the following MTL method:

$$\min_{\boldsymbol{w}_0, W} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|W\|_{2,2} \right\}. \tag{2.11}$$

The second term is an $L_{2,2}$-norm regularization. This term also groups the j-th variable by the L_2-norm. This is a well-known method as group lasso (Yuan and Lin, 2006), which estimates all variables in a group to be zero, when the value of L_2-norm for the group is below a certain threshold value. Consequently, this formulation induces the column-wise sparsity concerning W, which is similar to Problem (2.10). These sparse approaches have been extended in many ways to address various situations.

2.4 Clustering approach

The low-rank and the sparse approaches assume that all tasks are related to or have the same structure. However, in a practical situation, tasks with different characteristics

may be included in the task set, whose existence is unknown before the analysis. Imposing the shared structure to them can be detrimental in the contaminated situation. One solution is to group the task sets into those with a common structure, which can be performed by clustering the tasks based on their parameters. In this thesis. we mainly focus on this approach via fused regularization.

First, we describe the k-means method, which is well-known as a typical clustering method. Suppose that p-dimensional n data points $\{\boldsymbol{x}_i; i = 1, \ldots, n\}$ are observed. We predefine the number of groups C and attempt to classify these samples into distinctive C homogeneous groups. Each group is summarized into a center of the group $\boldsymbol{\mu}_c$ ($c = 1, \ldots, C$) called a centroid. Each observation is assigned to the group with the closest centroid.

To find the assignments of n samples into C groups, the procedure of k-means method optimize the following minimization problem:

$$\min_{\mu_c, \mathcal{I}_c, c=1,\ldots,C} \sum_{c=1}^{C} \sum_{i \in \mathcal{I}_c} \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|_2^2, \tag{2.12}$$

where $\mathcal{I}_c$ is a set of sample's index that belongs to c-th cluster. The standard algorithm of k-means method alternates the update of sample assignment $\mathcal{I}_c$ ($c = 1, \ldots, C$) and centroids $\boldsymbol{\mu}_c$ ($c = 1, \ldots, C$). The optimal $\boldsymbol{\mu}_c$ with fixed $\mathcal{I}_c$ is given by $\boldsymbol{\mu}_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \boldsymbol{x}_i$. The optimal $\mathcal{I}_c$ is given by assigning the sample i into $\mathcal{I}_c$ with the closest $\boldsymbol{\mu}_c$. This procedure is guaranteed to find a local minimum of Problem (2.12). However, the solution highly depends on the initial assignments and the number of possible combinations grows exponentially with the number of samples. Therefore, reasonable clustering results may not always be obtained.

Next, we consider grouping T tasks into C groups with their characteristics. The following MTL method based on the k-means (Zhou et al., 2011a) would be the most basic formulation for this purpose:

$$\min_{\substack{\boldsymbol{w}_0, W \\ \mu_c, \mathcal{I}_c, c=1,\ldots,C}} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \sum_{c=1}^{C} \sum_{m \in \mathcal{I}_c} \|\boldsymbol{w}_m - \boldsymbol{\mu}_c\|_2^2 \right\}. \tag{2.13}$$

Here, $\mathcal{I}_c$ is a set of task's index that belongs to m-th cluster. The second term is derived from the objective function of the k-means method (2.12). The alternating procedure is considered to optimize this problem. The updates of $(\boldsymbol{w}_0, W)$ with fixed $(\mathcal{I}_c, \boldsymbol{\mu}_c)$ are given by optimizing the following minimization problem:

$$\min_{w_{m0}, \boldsymbol{w}_m} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \|\boldsymbol{w}_m - \boldsymbol{\mu}_c\|_2^2 \right\}, \quad m \in \mathcal{I}_c,\ c = 1, \ldots, C.$$

Then, the problem is considered as ridge regression whose center is given by the task's centroid $\boldsymbol{\mu}_c$. The updates of $(\mathcal{I}_c, \boldsymbol{\mu}_c)$ with fixed $(\boldsymbol{w}_0, W)$ are given by the same procedure of k-means methods by replacing a sample $\boldsymbol{x}_i$ with a task's parameter $\boldsymbol{w}_m$. Therefore, each task's parameters are estimated around its centroid, and those centroids are jointly estimated based on the k-means method. In practice, this model may be rarely used. The reason may be that the alternating estimation algorithm in this model does not converge in most cases. However, it has interesting relationships with other multi-task learning methods, as we will see later.

2.4.1 Multi-task learning based on fused regularization

In the context of the clustering approach, there are many studies based on the fused regularization approach (e.g. Yamada et al. (2017); He et al. (2019); Dondelinger et al. (2020)), which takes the form:

$$\min_{\boldsymbol{w}_0, W} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \sum_{(m_1, m_2) \in \mathcal{E}} r_{m_1, m_2} \|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_q^l \right\}, \tag{2.14}$$

where r_{m_1,m_2} is a non-negative weight between m_1-th and m_2-th task and, $\mathcal{E}$ is a set of task pairs (m_1, m_2). The second term is a group fused L_q^l-norm regularization term. Since the values in that norm are estimated as smaller, the differences between regression coefficient vectors are shrunk. Consequently, this second term encourages similarity between tasks. In particular, when $l \leq 1$, the difference is estimated to be an exactly zero vector with a certain λ. Hence, tasks that have similar characteristics are estimated

to be $\boldsymbol{w}_{m_1} = \boldsymbol{w}_{m_2}$. Moreover, if $l = 1$ and $q \geq 1$, Problem (2.14) becomes a convex optimization problem. Then, a global minimum that leads to a simple interpretation of the estimated regression coefficients can be obtained. From those advanteges, Yamada et al. (2017) and He et al. (2019) used the setting $(l = 1, q = 2)$, while Dondelinger et al. (2020) and Zhang et al. (2024) used the setting $(l = 1, q = 1)$. The difference between these norms is that $(l = 1, q = 1)$ tends to fuse variables at the rather feature level, while $(l = 1, q = 2)$ fuses variables at the task level. The numerical experiments in He et al. (2019) showed that the fused L_2-norm regularization outperforms the fused L_1-norm regularization in almost all cases. This fused L_2-norm regularization is also called the network lasso (Hallac et al., 2015), which will be compared with our proposed method in Chapter 4.

In some studies (e.g. Hallac et al. (2015); Zhang et al. (2024)), it is supposed that the relationships between tasks are given as a graph structure $(\mathcal{V}, \mathcal{E}, R)$, and the fused regularization is used as a method to incorporate the graph information. Here, $\mathcal{V}$ is a set of the vertex corresponding to the task, and R is a $T \times T$ adjacency matrix whose element is given by

$$(R)_{m_1 m_2} = \begin{cases} r_{m_1,m_2} & (m_1, m_2) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, some extensions of Problem (2.14) are conducted in previous works. Those introductions, including a comparison with our work, are provided in Chapter 4.

2.5 Decomposition and robust approach

To address the heterogeneous structure among tasks, the clustering approach attempts to group tasks according to their characteristics. On the other hand, some MTL methods consider existing outlier tasks that have unique characteristics. To detect those tasks and reduce their influence on the shared structure, they decompose the param-

eters into commonly shared structures and task-specific structures. Those approaches are referred to as robust approach or decomposition approach.

Chen et al. (2011) firstly proposed a robust approach method referred to as robust multi-task learning (RMTL):

$$\min_{\boldsymbol{w}_0,W,U,V} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 \|U\|_* + \lambda_2 \|V\|_{1,2} \right\},$$
$$\text{s.t. } W = U + V,$$

where λ_1 and λ_2 are regularization parameters with non-negative values. In this method, the regression coefficient matrix W is decomposed into two parameter matrices $U \in \mathbb{R}^{T\times p}$ and $V \in \mathbb{R}^{T\times p}$. Furthermore, low-rank structure is induced to U, and row-wise sparsity is induced to V. From those regularizations, the main shared low-rank structure is imposed to U. If m-th task is an outlier task with task-specific characteristics, then m-th row of V denoted as $\boldsymbol{v}_m$ is estimated to be a non-zero vector. Thus, this method can viewed as a robust version of the low-rank method (2.9).

Gong et al. (2012) proposed a similar method referred to as Robust Multi-Task Feature Learning (rMTFL):

$$\min_{\boldsymbol{w}_0,W,U,V} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 \|U\|_{2,2} + \lambda_2 \|V\|_{1,2} \right\},$$
$$\text{s.t. } W = U + V,$$

From the second term, this method is a robust version of the sparse approach method (2.11). However, the regression coefficient vectors $\boldsymbol{w}_m$ on the outlier tasks do not show any sparsity pattern, which may be too restrictive. On the other hand, Jalali et al. (2010) proposed a sparse decomposition approach referred to as the dirty model:

$$\min_{\boldsymbol{w}_0,W,U,V} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 \|U\|_{2,\infty} + \lambda_2 \|V\|_{1,1} \right\}, \tag{2.15}$$
$$\text{s.t. } W = U + V,$$

The method imposes both column-wise sparsity by $L_{2,\infty}$-norm and element-wise sparsity by $L_{1,1}$-norm. Then, only when u_{mj} and v_{mj} are simultaneously estimated to be zero value, the w_{mj} becomes zero value. Jalali et al. (2010) shows the theoretical guarantee that the dirty model (2.15) outperforms Problem (2.10) under some conditions.

To perform clustering and obtain robustness against outlier tasks, Yao et al. (2019) proposed robust clustered multi-task learning (RCMTL):

$$\min_{\boldsymbol{w}_0, W, \Upsilon} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 \|W\|_F^2 + \lambda_2 \|W - \Upsilon W\|_{1,2} + \lambda_3 \|\Upsilon\|_{2,2} \right\},$$

where $\Upsilon = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T)^\top \in \mathbb{R}^{T \times T}$ is the coefficient matrix that describes the correlation between tasks. By the third term, $\boldsymbol{w}_m \simeq W^\top \boldsymbol{v}_m$ $(m = 1, \ldots, T)$ is facilitated, which means that m-th task is represented by the sum of all task's regression coefficient vector with the mixing weight $\boldsymbol{v}_m$. The third term induces column-wise sparsity for Υ, limiting the number of tasks representing the other task's regression coefficients. They claimed to have used the L_2^1-norm instead of the L_2^2-norm for the third term to provide robustness for outlier tasks. However, the effect of using the L_2^1 norm is to only make $\boldsymbol{w}_m$ and $W^\top \boldsymbol{v}_m$ exactly equal, and it is unclear why it would be robust to outlier tasks. They provided little explicit discussion of outlier tasks and their robustness.

2.6 Relationship between different approaches

We have presented some MTL approaches with different objectives. We introduce the relationship between some of the approaches.

First, we show the relationship between the low-rank approach (2.6) and the clustering approach based on the k-means method (2.13). Let the regularization terms in Problem (2.13) be $\Omega_{\mathrm{k}}(W)$. From some algebra, the $\Omega_{\mathrm{k}}(W)$ can be written as

$$\Omega_{\mathrm{k}}(W) = \lambda(\|W\|_F^2 - \mathrm{tr}(W^\top \Xi \Xi^\top W)), \tag{2.16}$$

where Ξ is a $T \times C$ orthogonal cluster assignment matrix whose element is defined by

$$(\Xi)_{mc} = \begin{cases} \frac{1}{\sqrt{n_m}} & m \in \mathcal{I}_c, \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

We consider dropping the specific structure of Ξ in (2.17) while maintaining the orthogonality. Then, the relaxed minimization problem of (2.16) based on k-means is equivalent to

$$\max_{\Xi} \operatorname{tr}(\Xi^\top W W^\top \Xi), \quad \text{s.t. } \Xi^\top \Xi = I_C. \tag{2.18}$$

Therefore, the relaxed k-means is the principal component analysis for $W^\top$, which characterizes the difference between the estimation of Γ in (2.8) and Ξ in (2.18). In a word, the low-rank approach uses information concerning the correlation of task variables, while the clustering approach uses similarity between tasks. Moreover, the MTL method via original k-means requires a more limited structure than the low-rank approach, which may make its estimation more difficult.

Next, we show the relevance between the relaxed k-means and the fused regularization approach (2.14). We consider the setting $(l = 2, q = 2)$ for Problem (2.14), and let $\Omega_{\mathrm{FL}}(W)$ be the regularization term. The regularization term can be written as follows:

$$\begin{aligned} \Omega_{\mathrm{FL}}(W) &= \lambda \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_2^2 \\ &= 2\lambda \operatorname{tr}(W^\top L W), \end{aligned}$$

where L is a graph Laplacian matrix given by $L = D - R$, and D is a $T \times T$ diagonal degree matrix with diagonal component $(D)_{mm} = d_m = \sum_{m_2=1}^{T} r_{m,m_2}$. From the formulation, the $\Omega_{\mathrm{FL}}(W)$ is also known as a Laplacian regularization. Furthermore, using a normalized graph Laplacian calculated by

$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I_T - D^{-\frac{1}{2}} R D^{-\frac{1}{2}},$$

leads to the normalized Laplacian regularization:

$$\sum_{(m_1,m_2)\in\mathcal{E}} \frac{r_{m_1,m_2}}{\sqrt{d_{m_1} d_{m_2}}} \|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_F^2 = 2\operatorname{tr}(W^\top (I_T - D^{-\frac{1}{2}} R D^{-\frac{1}{2}}) W),$$

By comparing the normalized Laplacian regularization with Problem (2.16), we can find that the difference is only on the matrix $\Xi\Xi^\top$ and $D^{-\frac{1}{2}}RD^{-\frac{1}{2}}$. Thus, the normalized Laplacian regularization can be viewed as a special case of the relaxed k-means approach method. Because the matrix $D^{-\frac{1}{2}}RD^{-\frac{1}{2}}$ is fixed through the optimization and the elements are non-negative, the regularization term may appear to be more restrictive than relaxed k-means.

Next, we give the connection between the Laplacian regularization ($l = 2, q = 2$) and the setting ($l = 1, q = 2$) also known as the network lasso (Hallac et al., 2015) based on the majorization – minimization (MM) algorithm (e.g. Lange (2016)).

The MM algorithm performs alternately minimizing the majorization function $\varphi(\cdot, \boldsymbol{w})$ instead of the objective function $\Omega(\boldsymbol{w})$. The majorization function $\varphi(\cdot, \boldsymbol{r})$ is defined to satisfy the following relationship:

$$\begin{aligned} \Omega(\boldsymbol{w}) &= \varphi(\boldsymbol{w}, \boldsymbol{w}), \\ \Omega(\boldsymbol{w}) &\leq \varphi(\boldsymbol{w}, \boldsymbol{\tau}), \end{aligned} \tag{2.19}$$

for every $\boldsymbol{w}, \boldsymbol{\tau}$. In the MM algorithm, the updates given by

$$\boldsymbol{w}^{(t)} = \arg\min_{\boldsymbol{w}} \varphi(\boldsymbol{w}, \boldsymbol{w}^{(t-1)}).$$

From the relationship (2.19), this update procedure satisfies the following descent property:

$$\Omega(\boldsymbol{w}^{(t)}) \leq \varphi(\boldsymbol{w}^{(t)}, \boldsymbol{w}^{(t-1)}) \leq \varphi(\boldsymbol{w}^{(t-1)}, \boldsymbol{w}^{(t-1)}) \leq \Omega(\boldsymbol{w}^{(t-1)})$$

Here, the superscript t with brackets represents the number of iterations in the update procedure. Thus, the minimization of the objective function is performed by iteratively minimizing the majorization function. On the other hand, for a concave function $f(\cdot)$, we have

$$f(\boldsymbol{w}) \leq f(\boldsymbol{\tau}) + \frac{\partial}{\partial \boldsymbol{w}} f(\boldsymbol{\tau})^\top (\boldsymbol{w} - \boldsymbol{\tau}).$$

Since the right-hand side satisfies the relationship (2.19), we can set the majorization

function as

$$\varphi(\boldsymbol{w}, \boldsymbol{\tau}) = f(\boldsymbol{\tau}) + \frac{\partial}{\partial \boldsymbol{w}} f(\boldsymbol{\tau})^\top (\boldsymbol{w} - \boldsymbol{\tau}).$$

Thus, for the concave function $\sqrt{\cdot}$ without origin, we obtain majorization function

$$\varphi(w, \tau) = \frac{\sqrt{\tau}}{2} + \frac{1}{2\sqrt{\tau}} w.$$

Because the original objective function of the network lasso without weights r_{m_1,m_2} is given by $\Omega(W) = \sum_{(m_1,m_2)\in\mathcal{E}} \sqrt{\|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_2^2}$, its majorization function becomes

$$\varphi(W, \tilde{R}) = \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \tilde{r}_{m_1,m_2} \|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_2^2 + \frac{1}{4\tilde{r}_{m_1,m_2}} \right\}.$$

Here, we set $\frac{1}{2\sqrt{\tau_{m_1,m_2}}}$ as $\tilde{r}_{m_1,m_2}$. The optimal $\tilde{R}$ is given by $\tilde{r}^{(t)}_{(m_1,m_2)} = \frac{1}{\left\|\boldsymbol{w}_{m_1}^{(t-1)} - \boldsymbol{w}_{m_2}^{(t-1)}\right\|_2}$. Consequently, the MTL method based on the network lasso is equivalent to the following joint optimization problem concerning W and $\tilde{R}$:

$$\min_{\boldsymbol{w}_0, W, \tilde{R}} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda \left(\sum \tilde{r}_{m_1,m_2} \|\boldsymbol{w}_{m_1} - \boldsymbol{w}_{m_2}\|_2^2 + \frac{1}{4\tilde{r}_{m_1,m_2}} \right) \right\}.$$

Thus, network lasso can be viewed as an extension of the graph Laplacian regularization in which the estimation of R is adapted to the data. Thus, it is also related to the relaxed k-means method. However, the last term avoids the value of $\tilde{r}_{m_1,m_2}$ being close to zero. In contrast to the original k-means MTL, the network lasso does not eliminate interference between tasks belonging to different clusters, probably due to the convex relaxation. Although He et al. (2019) proposed a scalable optimization algorithm for the MTL method with the network lasso using the same method described above, they did not mention the connection to the MM algorithm.

Chapter 3

Convex clustering

Convex clustering is a fundamental technique that provides a convex relaxation of k-means and hierarchical clustering methods. It enables stable cluster estimation by replacing the discrete cluster assignment in k-means with continuous centroid parameters. This approach plays a central role in our proposed methods in the subsequent chapters, where we extend and adapt its framework to multi-task learning settings. This chapter describes the formulation of convex clustering and presents an efficient estimation algorithm based on the modified alternating direction method of multipliers.

3.1 Formulation

Suppose that we have n observed p-dimensional data $\{\boldsymbol{x}_i; i = 1, \ldots, n\}$. Convex clustering (Pelckmans et al. (2005); Hocking et al. (2011); Lindsten et al. (2011)) classifies these data into exclusive clusters as a convex optimization problem and is formulated as follows:

$$\min_{U} \left\{ \sum_{i=1}^{n} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{u}_i\|_2^2 + \lambda \sum_{(i_1,i_2)\in\mathcal{E}} r_{i_1,i_2} \|\boldsymbol{u}_{i_1} - \boldsymbol{u}_{i_2}\|_q \right\}, \tag{3.1}$$

where $\boldsymbol{u}_i \in \mathbb{R}^p$ is a centroid vector for i-th sample and $U = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)^\top \in \mathbb{R}^{n\times p}$. Unlike k-means clustering where a fixed number of centroids is predetermined, convex

clustering assigns a centroid to each data point. The first term is a loss function that measures the fidelity between data points and their corresponding centroid vectors. The second term is a fused regularization term encouraging the fusion of centroids. For the second term, $q = 1, 2$, or ∞ is typically used to shrink the difference between $\boldsymbol{u}_{i_1}$ and $\boldsymbol{u}_{i_2}$ into exactly zero. When the value of $\boldsymbol{u}_{i_1}$ and $\boldsymbol{u}_{i_2}$ are estimated to be the same, corresponding samples $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$ are considered as belonging to the same cluster.

Tan and Witten (2015) established a connection between convex clustering and k-means method. Setting $q = 0$ and uniform weights $r_{m_1,m_2} = 1$ yields:

$$\min_{U} \left\{ \sum_{i=1}^{n} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{u}_i\|_2^2 + \lambda \sum_{(i_1,i_2)\in\mathcal{E}} \mathbb{I}(\boldsymbol{u}_{i_1} = \boldsymbol{u}_{i_2}) \right\}.$$

Because the second term separates the estimation of $\boldsymbol{u}_i$ with samples from different clusters, we can rewrite the problem as

$$\min_{\mu_c,\mathcal{I}_c,c=1,\ldots,C} \left\{ \sum_{c=1}^{C} \sum_{i\in\mathcal{I}_c} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|_2^2 + \lambda \sum_{(i_1,i_2)\in\mathcal{E}} \sum_{c=1}^{C} \mathbb{I}(i_1 \in \mathcal{I}_c, i_2 \notin \mathcal{I}_c) \right\},$$

where $\boldsymbol{\mu}_c$ and $\mathcal{I}_c$ have the same definition in the k-means method (2.12). The convex clustering with $q = 0$ has the same objective function up to the first term. The second term penalizes the number of samples from the different clusters, which probably induces unbalanced clusters, unlike the k-means. Thus, convex clustering is a convex relaxation almost the same problem as k-means. In other words, a centroid $\boldsymbol{u}_i$ is considered as a biased centroid in the k-means for $q \neq 0$, which means that the estimated centroids $\widehat{\boldsymbol{u}}_i$ are affected by shrinkage with other cluster's centroids.

The value of weights r_{i_1,i_2} in (3.1) are often calculated by the combination of k-nearest neighbor and the Gaussian kernel as follows:

$$r_{i_1,i_2} = \begin{cases} \exp\left(-\alpha \|\boldsymbol{x}_{i_1} - \boldsymbol{x}_{i_2}\|_2^2\right) & \boldsymbol{x}_{i_1} \text{ is a } k\text{-nearest neighbor of } \boldsymbol{x}_{i_2} \text{ or vice versa,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where α (≥ 0) is a tuning parameter. Lindsten et al. (2011) used only k-nearest neighbor ($\alpha = 0$), while other studies (e.g. Pelckmans et al. (2005), Chi and Lange

(2015)) incorporated the Gaussian kernel ($\alpha > 0$). Those weights enable the reduction of the shrinkage between distant samples. It has been empirically confirmed that those weights improve the clustering performance. Moreover, setting many weights to zero can also reduce computational costs.

3.2 Estimation algorithm

Chi and Lange (2015) proposed an estimation algorithm of the convex clustering based on the alternating direction methods of multipliers (ADMM). The algorithm was further improved to reduce the computational time by Shimmura and Suzuki (2022).

Let A be a $p \times n$ matrix, and let $f\colon \mathbb{R}^n \to \mathbb{R}$ and $h\colon \mathbb{R}^m \to \mathbb{R}$ be convex functions. The ADMM is used for solving the minimization problem in terms of $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ taking the form:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \left\{ f(\boldsymbol{x}) + h(\boldsymbol{y}) \right\}, \quad \text{s.t.} \quad A\boldsymbol{x} = \boldsymbol{y}.$$

The augmented Lagrangian for the constraint problem is expressed as

$$L_\nu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{s}^\top (A\boldsymbol{x} - \boldsymbol{y}) + \frac{\nu}{2} \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2,$$

where $\boldsymbol{s}$ is a p-dimensional vector of Lagrangian multipliers and ν (≥ 0) is a tuning parameter. To solve the problem, the standard ADMM algorithm iterates the following updates until convergence:

$$\begin{aligned} \boldsymbol{x}^{(t+1)} &= \arg\min_{\boldsymbol{x}} L_\nu(\boldsymbol{x}, \boldsymbol{y}^{(t)}, \boldsymbol{s}^{(t)}), \\ \boldsymbol{y}^{(t+1)} &= \arg\min_{\boldsymbol{y}} L_\nu(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}, \boldsymbol{s}^{(t)}), \\ \boldsymbol{s}^{(t+1)} &= \boldsymbol{s}^{(t)} + \nu(A\boldsymbol{x}^{(t+1)} - \boldsymbol{y}^{(t+1)}). \end{aligned} \tag{3.3}$$

For the updates (3.3), Shimmura and Suzuki (2022) considered replacing the separated updates in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$ with the following joint update:

$$(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^{(t+1)}) = \arg\min_{\boldsymbol{x},\boldsymbol{y}} \left\{ L_\nu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}^{(t)}) \right\}.$$

If we consider updating only $\boldsymbol{x}$ by the joint optimization, the update can be written as

$$\begin{aligned}\boldsymbol{x}^{(t+1)} &= \arg\min_{\boldsymbol{x}} \left\{ \min_{\boldsymbol{y}} L_\nu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}^{(t)}) \right\} \\ &= \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \min_{\boldsymbol{y}} \left\{ h(\boldsymbol{y}) + \boldsymbol{s}^{\top(t)}(A\boldsymbol{x} - \boldsymbol{y}) + \frac{\nu}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 \right\} \right\} \\ &= \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) - r^*(\nu A\boldsymbol{x} + s^{(t)}) + \boldsymbol{s}^{\top(t)} A\boldsymbol{x} + \frac{\nu}{2}\|A\boldsymbol{x}\|_2^2 \right\},\end{aligned} \tag{3.4}$$

where $r^*(\cdot)$ is the conjugate function of $r(\boldsymbol{z}) = h(\boldsymbol{z}) + \frac{\nu}{2}\|\boldsymbol{z}\|_2^2$. This indicates that we can update $\boldsymbol{x}$ without updating $\boldsymbol{y}$. Furthermore, the following lemma motivates us to solve the minimization problem (3.4) by the gradient method.

Lemma 1 (Shimmura and Suzuki (2022); Theorem 1)**.** *If we define* $\phi_1(\boldsymbol{x}) = f(\boldsymbol{x}) + \min_{\boldsymbol{y}}(h(\boldsymbol{y}) + \boldsymbol{s}^{\top(t)}(A\boldsymbol{x} - \boldsymbol{y}) + \frac{\nu}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2)$*,* ϕ_1 *is differentiable. Furthermore, we have*

$$\frac{\partial}{\partial \boldsymbol{x}}\phi_1(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}) + A^\top(\mathrm{prox}_{\nu h^*}(\nu A\boldsymbol{x} + \boldsymbol{s}^{(t)})),$$

where $\mathrm{prox}_g(\cdot)$ *is the proximal map of* $g(\cdot)$ *defined as*

$$\mathrm{prox}_g(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \left\{ g(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \right\}.$$

Thus, the update of $\boldsymbol{x}$ is given by the convergence point of the gradient method. Here, the update in the gradient method is given by

$$\boldsymbol{x}^{(l+1)} = \boldsymbol{x}^{(l)} - \iota \left\{ \frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}^{(l)}} + A^\top(\mathrm{prox}_{\nu h^*}(\nu A\boldsymbol{x}^{(l)} + \boldsymbol{s}^{(t)})) \right\},$$

where ι is a step size with a non-negative value. Shimmura and Suzuki (2022) considered accelerating the convergence rate of the gradient method based on Nesterov's acceleration method (Nesterov, 1983). To guarantee the convergence, the accelerated gradient method requires setting $\iota \leq 1/L_{\mathrm{c}}$ for $L_{\mathrm{c}} > 0$ such that

$$\left\| \frac{\partial}{\partial \boldsymbol{x}}\phi_1(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_1} - \frac{\partial}{\partial \boldsymbol{x}}\phi_1(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_2} \right\|_2 \leq L_{\mathrm{c}}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2.$$

The update of $\boldsymbol{s}$ via ADMM can be converted by Moreau decomposition as

$$\begin{aligned}\boldsymbol{s}^{(t+1)} &= \boldsymbol{s}^{(t)} + \nu(A\boldsymbol{x}^{(t+1)} - \boldsymbol{y}^{(t+1)}) \\ &= \mathrm{prox}_{\nu h^*}(\boldsymbol{s}^{(t)} + \nu A\boldsymbol{x}^{(t+1)}).\end{aligned} \tag{3.5}$$

Consequently, the updates of the ADMM algorithm based on the joint updates are given by

$$\boldsymbol{x}^{(t+1)} = \mathrm{AGrad}(\boldsymbol{x}^{(t)}, \boldsymbol{s}^{(t)}),$$
$$\boldsymbol{s}^{(t+1)} = \mathrm{prox}_{\nu h^*}(\boldsymbol{s}^{(t)} + \nu A\boldsymbol{x}^{(t+1)}),$$

where $\mathrm{AGrad}(\cdot,\cdot)$ is a function returning the convergence point computed by the accelerated gradient method. Thus, we do not require any updates of $\boldsymbol{y}$. We refer to this algorithm as the modified ADMM.

To solve the convex clustering by the modified ADMM, Problem (3.1) is equivalently rewritten as

$$\min_{U,B} \frac{1}{2}\|X - U\|_F^2 + \lambda \sum_{(i_1,i_2)\in\mathcal{E}} r_{i_1,i_2}\|\boldsymbol{b}_{(i_1,i_2)}\|_2,$$
$$\text{s.t. } B - A_{\mathcal{E}}U = 0,$$

where B is a $|\mathcal{E}| \times p$ matrix whose row vector is $\boldsymbol{b}_{(i_1,i_2)} \in \mathbb{R}^p$ and $A_{\mathcal{E}}$ is a $|\mathcal{E}| \times T$ matrix whose row vector $\boldsymbol{a}_{(i_1,i_2)}$ is defined as

$$(\boldsymbol{a}_{(i_1,i_2)})_i = \begin{cases} 1 & i = i_1, \\ -1 & i = i_2, \\ 0 & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n. \tag{3.6}$$

Note that we consider only the L_2-norm for the shrinkage of the centroids. Then, the augmented Lagrangian is given by

$$L_\nu(U, B, S) = \frac{1}{2}\|X - U\|_F^2 + \lambda \sum_{(i_1,i_2)\in\mathcal{E}} r_{m_1,m_2}\|\boldsymbol{b}_{(m_1,m_2)}\|_2 +$$
$$+ \mathrm{tr}(S^\top(B - A_{\mathcal{E}}U)) + \frac{\nu}{2}\|B - A_{\mathcal{E}}U\|_F^2,$$

where S is a $|\mathcal{E}| \times p$ Lagrangian multipliers matrix whose row vector is $\boldsymbol{s}_{(i_1,i_2)} \in \mathbb{R}^p$. From Lemma 1, the update of inner gradient method is given by

$$U^{(l+1)} = U^{(l)} - \iota\{\lambda(U^{(l)} - X)) + A_{\mathcal{E}}^\top F\}, \tag{3.7}$$

where F is the matrix whose each row is updated by

$$F_{(i_1,i_2)} = \mathrm{prox}((S^{(t)} + \nu A_{\mathcal{E}} \cdot C^{(l)})_{(i_1,i_2)}, \lambda_2 r_{i_1,i_2}),$$

where $\mathrm{prox}(\cdot, \lambda)$ is defined as

$$\mathrm{prox}(\boldsymbol{z}, \lambda) = \min(\|\boldsymbol{z}\|_2, \lambda)\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}. \tag{3.8}$$

Eq. (3.8) is a proximal map of h^* derived from $h(\boldsymbol{z}) = \lambda\|\boldsymbol{z}\|_2$. Shimmura and Suzuki (2022) showed that $\iota = \frac{1}{1+2\nu \max_{i=1,\ldots,n} A_{\mathcal{E}}^\top A_{\mathcal{E}}}$ satisfies the condition $\iota \leq \frac{1}{L_c}$. By combining the inner update (3.7) and accelerated gradient method, we obtain an update of $U^{(t+1)}$ from the convergence point. Similarly, from the update (3.5), the update of S for the convex clustering is given by

$$S^{(t+1)}_{(i_1,i_2)} = \mathrm{prox}((S^{(t)} + \nu A_{\mathcal{E}} \cdot U^{(t+1)})_{(i_1,i_2)}, \lambda_2 r_{i_1,i_2}), \quad \text{for } (i_1, i_2) \in \mathcal{E}.$$

These estimation procedures are summarized into Algorithm 1:

Algorithm 1 Estimation algorithm for the convex clustering via the modified ADMM

function CVX(X, R, λ)

Initialize; $U^{(0)} = X$

Calculate $A_{\mathcal{E}}$ by Eq. (3.6) from R

$L = A_{\mathcal{E}}^{\top} A_{\mathcal{E}}, \iota = \frac{1}{1+2\max_{i=1,\ldots,n}((L)_{ii})}$

while until convergence of $U^{(t)}$ **do**

$l = 0, \alpha^{(0)} = 1, H^{(0)} = U^{(t)}, E^{(0)} = U^{(t)}$

while until convergence of $H^{(l)}$ **do**

for $(i_1, i_2) \in \mathcal{E}$ **do**

$F_{(i_1,i_2)} = \mathrm{prox}((S^{(t)} + \nu A_{\mathcal{E}} E^{(t)})_{(i_1,i_2)}, \lambda_2 r_{i_1,i_2})$

end for

$H^{(l+1)} = E^{(l)} - \iota\{\lambda(E^{(l)} - X) + A_{\mathcal{E}}^{\top} F\}$

$\alpha^{(l+1)} = \frac{1+\sqrt{1+4(\alpha^{(l)})^2}}{2}$

$E^{(l+1)} = E^{(l)} + \frac{\alpha^{(l)}-1}{\alpha^{(l+1)}}(H^{(l+1)} - H^{(l)})$

end while

$U^{(t+1)} = H^{(l)}$

for $(i_1, i_2) \in \mathcal{E}$ **do**

$S_{(i_1,i_2)}^{(t+1)} = \mathrm{prox}((S_{(i_1,i_2)}^{(t)} + \nu A_{\mathcal{E}} U^{(t+1)})_{(i_1,i_2)}, \lambda r_{i_1,i_2})$

end for

end while

Output: U

end function

Chapter 4

Multi-task learning with separated parameters for regression and task fusion

Multi-task learning methods based on clustering would be a natural option in the practical analysis, since the tasks are obtained from heterogeneous environments and may have different characteristics. However, the problem of clustering methods depending on initial values is widely known, as with the k-means method and the finite Gaussian mixture model. Therefore, it is difficult to employ k-means for clustering tasks due to its instability. Indeed, we have observed that the estimation algorithm of the MTL method based on k-means does not converge to the stationary point in almost all cases. Therefore, it may be the reason why many studies of clustering approach MTLs have employed the fused regularization. By using the fused regularization, the discontinuity of k-means in estimating the cluster assignment can at least be removed, or it can be formulated as a convex optimization problem. In this chapter, we propose the MTL method that reduces the disadvantage of the fused regularization terms.

4.1 Proposed method

4.1.1 Multi-task learning via convex clustering

The fused regularization term has the problem of a task being affected by other tasks belonging to other clusters whenever their weight is non-zero values. To address this issue, Yamada et al. (2017) and He et al. (2019) calculated the weights r_{m_1,m_2} using k-nearest neighbor, which may be based on the convex clustering. Zhou and Zhao (2016) and Shimamura and Kawano (2021) proposed the methods that treat the weights as latent parameters and estimate them and regression coefficient parameters simultaneously. Because the latter approach induces the non-convexity of the model, it is difficult to construct the estimation algorithm converging into the global minimum.

To overcome this problem, we propose the following minimization problem:

$$\min_{\boldsymbol{w}_0,W,U} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \|\boldsymbol{w}_m - \boldsymbol{u}_m\|_2^2 + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \right\}, \tag{4.1}$$

where $\boldsymbol{u}_m \in \mathbb{R}^p$ is a centroid for m-th task, $U = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T)^\top$ is a $T \times p$ matrix, λ_1 and λ_2 are tuning parameters with non-negative values. The second term is a squared-L_2 norm to estimate the value of $\boldsymbol{w}_m$ around that of $\boldsymbol{u}_m$. The third term is a L_2-norm to perform the clustering of $\boldsymbol{u}_m$.

In Problem (4.1), the regression coefficient vectors $\boldsymbol{w}_m$ are not shrunk directly unlike Problem (2.14), while $\boldsymbol{u}_m$ are shrunk and clustered. When the value of λ_1 is large, $\boldsymbol{w}_m$ is estimated to be the same value of $\boldsymbol{u}_m$, which is close to Problem (2.14). However, when the value of λ_1 is small, the value of $\boldsymbol{w}_m$ can be estimated to be different from that of $\boldsymbol{u}_m$. Therefore, we can expect to reduce the shrinkage among irrelevant tasks. Because the second and third terms are viewed as regularization terms derived from the model of convex clustering, we refer to this model as MTLCVX (**M**ulti-**T**ask **L**earning via **C**on**V**e**X** clustering).

We set the weights r_{m_1,m_2} in (4.1) as in Yamada et al. (2017):

$$R = \frac{S^\top + S}{2}, \quad (S)_{m_1 m_2} = \begin{cases} 1 & \widehat{\boldsymbol{w}}_{m_1}^{\mathrm{STL}} \text{ is a } k\text{-nearest neighbor of } \widehat{\boldsymbol{w}}_{m_2}^{\mathrm{STL}}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.2}$$

where R is a $T \times T$ matrix whose each component is $(R)_{m_1 m_2} = r_{m_1,m_2}$ and $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ is an estimated regression coefficient vector for m-th task by single-task learning such as OLS, ridge, and lasso. From this equation, if m_1-th task and m_2-th task are k-nearest neighbors of each other, then $r_{m_1,m_2} = 1$. If they are k-nearest neighbors from only one side, then $r_{m_1,m_2} = 0.5$. He et al. (2019) only set $r_{m_1,m_2} = \{0, 1\}$ in a similar way. These constructed weights would be based on that of convex clustering (3.2), which is used to reduce the computational costs by setting some r_{m_1,m_2} to zero and to improve the estimation results of clustering.

In Problem (4.1), since the second and third terms introduce interactions between the parameters $\boldsymbol{w}_m$ and $\boldsymbol{u}_m$, the joint convexity of the objective function may not be obvious. In the following section, we prove that MTLCVX is indeed jointly convex with respect to $(\boldsymbol{w}_0, W)$ and U for both squared-loss and logistic-loss functions.

4.1.2 Convexity of MTLCVX

We show the joint convexity of MTLCVX with respect to $\boldsymbol{w}_m$ and $\boldsymbol{u}_m$. Because the sum of the convex functions is also a convex function, it suffices to show the joint convexity of the sum of the first and second terms. Then, we show the positive-semidefiniteness of the Hessian matrix concerning the sum of the first and second terms in Problem (4.1). Because the loss function and the regularization terms for m-th task are independent of those for other tasks, we omit the index for the number of tasks to simplify the notation.

4.1.2.1 For the squared-loss function

We consider the loss function in Eq. (2.3). Let $\boldsymbol{\theta} = (\boldsymbol{w}^\top, \boldsymbol{u}^\top)^\top$ be a parameter vector. We set $l(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{y} - X\boldsymbol{w}\|_2^2 + \frac{\lambda_1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|_2^2$. Then, the derivation of the Hessian matrix for $l(\boldsymbol{\theta})$ is as follows:

$$\begin{aligned}\frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} &= \begin{pmatrix} \frac{\partial^2 l}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} & \frac{\partial^2 l}{\partial \boldsymbol{w} \partial \boldsymbol{u}^\top} \\ \frac{\partial^2 l}{\partial \boldsymbol{u} \partial \boldsymbol{w}^\top} & \frac{\partial^2 l}{\partial \boldsymbol{u} \partial \boldsymbol{u}^\top} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{n} X^\top X + \lambda_1 I_p & -\lambda_1 I_p \\ -\lambda_1 I_p & \lambda_1 I_p \end{pmatrix}.\end{aligned}$$

Let $\boldsymbol{b} \in \mathbb{R}^p$ and $\boldsymbol{c} \in \mathbb{R}^p$ be non-zero vectors, and we set $\boldsymbol{a} = (\boldsymbol{b}^\top, \boldsymbol{c}^\top)^\top$. The quadratic form of the Hessian matrix is calculated as follows:

$$\begin{aligned}\boldsymbol{a}^\top \frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \boldsymbol{a} &= \begin{pmatrix} \boldsymbol{b}^\top & \boldsymbol{c}^\top \end{pmatrix} \begin{pmatrix} \frac{1}{n} X^\top X + \lambda_1 I_p & -\lambda_1 I_p \\ -\lambda_1 I_p & \lambda_1 I_p \end{pmatrix} \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{c} \end{pmatrix} \\ &= \boldsymbol{b}^\top (\frac{1}{n} X^\top X + \lambda_1 I_p) \boldsymbol{b} - 2\lambda_1 \boldsymbol{b}^\top \boldsymbol{c} + \lambda_1 \boldsymbol{c}^\top \boldsymbol{c} \\ &= \frac{1}{n}\|X\boldsymbol{b}\|_2^2 + \lambda_1 \|\boldsymbol{b} - \boldsymbol{c}\|_2^2 \geq 0.\end{aligned}$$

This means that $l(\boldsymbol{\theta})$ is a positive-semidefinite. Thus, the sum of the first and second terms in Problem (4.1) is a convex function when the squared loss function is used.

4.1.2.2 For the logistic-loss function

We consider the loss function in Eq. (2.4). Let $\boldsymbol{\theta} = (w_0, \boldsymbol{w}^\top, \boldsymbol{u}^\top)^\top$ be a parameter vector. We set $l(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^n \{y_i(w_0 + \boldsymbol{w}^\top)\boldsymbol{x}_i - \log(1 + \exp(w_0 + \boldsymbol{w}^\top \boldsymbol{x}_i))\} + \frac{\lambda_1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|_2^2$. Then, the derivation of the Hessian matrix for $l(\boldsymbol{\theta})$ is as follows:

$$\begin{aligned}\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} &= \begin{pmatrix} \frac{\partial^2 l}{\partial w_0^2} & \frac{\partial^2 l}{\partial w_0 \partial \boldsymbol{w}^\top} & \frac{\partial^2 l}{\partial w_0 \partial \boldsymbol{u}^\top} \\ \frac{\partial l^2}{\partial \boldsymbol{w} \partial w_0} & \frac{\partial^2 l}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top} & \frac{\partial^2 l}{\partial \boldsymbol{w} \partial \boldsymbol{u}^\top} \\ \frac{\partial^2 l}{\partial \boldsymbol{u} \partial w_0} & \frac{\partial^2 l}{\partial \boldsymbol{u} \partial \boldsymbol{w}^\top} & \frac{\partial^2 l}{\partial \boldsymbol{u} \partial \boldsymbol{u}^\top} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{n}\mathbf{1}_n^\top \Pi (I_n - \Pi)\mathbf{1}_n & \frac{1}{n}\mathbf{1}_n^\top \Pi (I_n - \Pi) X & \mathbf{0}^\top \\ \frac{1}{n} X^\top \Pi (I_n - \Pi)\mathbf{1}_n & \frac{1}{n} X^\top \Pi (I_n - \Pi) X + \lambda_1 I_n & -\lambda_1 I_n \\ \mathbf{0} & -\lambda_1 I_n & \lambda_1 I_n \end{pmatrix},\end{aligned}$$

where Π is an $n \times n$ diagonal matrix with a diagonal element $(\Pi)_{ii} = 1 - 1/(1+\exp(w_0 + \boldsymbol{w}^\top \boldsymbol{x}_i))$. Let $b \in \mathbb{R}, \boldsymbol{c} \in \mathbb{R}^p$, and $\boldsymbol{d} \in \mathbb{R}^p$ be a non-zero scalar and vectors, and we set $\boldsymbol{a} = (b, \boldsymbol{c}^\top, \boldsymbol{d}^\top)^\top$. The quadratic form of the Hessian matrix is calculated as follows:

$$
\begin{aligned}
\boldsymbol{a}^\top \frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \boldsymbol{a} &= \begin{pmatrix} b & \boldsymbol{c}^\top & \boldsymbol{d}^\top \end{pmatrix} \begin{pmatrix} \frac{1}{n}\mathbf{1}_n^\top \Pi(I_n - \Pi)\mathbf{1}_n & \frac{1}{n}\mathbf{1}_n^\top \Pi(I_n - \Pi)X & \mathbf{0}^\top \\ \frac{1}{n}X^\top \Pi(I_n - \Pi)\mathbf{1}_n & \frac{1}{n}X^\top \Pi(I_n - \Pi)X + \lambda_1 I_n & -\lambda_1 I_n \\ \mathbf{0} & -\lambda_1 I_n & \lambda_1 I_n \end{pmatrix} \begin{pmatrix} b \\ \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix} \\
&= \frac{b^2}{n}\mathbf{1}^\top \Pi(I_n - \Pi)\mathbf{1} + \frac{2b}{n}\mathbf{1}^\top \Pi(I_n - \Pi)X\boldsymbol{c} + \frac{1}{n}\boldsymbol{c}^\top X^\top \Pi(I_n - \Pi)X\boldsymbol{c} \\
&+ \lambda_1(\boldsymbol{c}^\top \boldsymbol{c} - 2\boldsymbol{c}^\top \boldsymbol{d} + \boldsymbol{d}^\top \boldsymbol{d}) \\
&= \frac{1}{n}\|(\Pi(I_n - \Pi))^{1/2}(X\boldsymbol{c} - b\mathbf{1})\|_2^2 + \lambda_1 \|\boldsymbol{c} - \boldsymbol{d}\|_2^2 \geq 0.
\end{aligned}
$$

This means that $l(\boldsymbol{\theta})$ is a convex function in terms of w_0, $\boldsymbol{w}$ and $\boldsymbol{u}$. Thus, the sum of the first and second terms in Problem (4.1) is a convex function when the logistic loss function is used.

The positive-semidefiniteness of the Hessian matrices in both cases establishes that MTLCVX is jointly convex regardless of whether we use the squared loss or logistic loss function. This convexity ensures the existence of a global minimum. Furthermore, MTLCVX serves as a direct convex relaxation of the k-means based method (2.13).

4.1.3 Multi-task learning via adaptive convex clustering

A drawback of Eq. (4.2) is that weights r_{m_1,m_2} may have some noises, since the estimated value $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ may not be accurate. To address it, we consider calculating weights r_{m_1,m_2} as in the adaptive lasso (Zou, 2006):

$$
\min_{\boldsymbol{w}_0, W, U} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \|\boldsymbol{w}_m - \boldsymbol{u}_m\|_2^2 + \lambda_2 \sum_{(m_1,m_2) \in \mathcal{E}} \widehat{r}_{m_1,m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \right\}, \tag{4.3}
$$

where $\widehat{r}_{m_1,m_2}$ is an adaptive weight. This weight is computed as follows:

$$\widehat{r}_{m_1,m_2} = \frac{1}{\|\widehat{\boldsymbol{u}}_{m_1}(\mathrm{MTLCVX}) - \widehat{\boldsymbol{u}}_{m_2}(\mathrm{MTLCVX})\|_2}\delta,$$
$$\delta = \left(\sum_{(m_1,m_2)\in\mathcal{E}} \frac{1}{\|\widehat{\boldsymbol{u}}_{m_1}(\mathrm{MTLCVX}) - \widehat{\boldsymbol{u}}_{m_2}(\mathrm{MTLCVX})\|_2}\right)^{-1} \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2},$$

where $\widehat{\boldsymbol{u}}_m(\mathrm{MTLCVX})$ is an estimated value of a centroid $\boldsymbol{u}_m$ in Problem (4.1), and δ is a scaling parameter. The scaling parameter δ is defined to ensure $\sum_{(m_1,m_2)\in\mathcal{E}} \widehat{r}_{(m_1,m_2)} = \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2}$. This scaling prevents large fluctuations in the value of the optimal regularization parameters empirically. We refer to Problem (4.3) as MTLACVX (**M**ulti-**T**ask **L**earning via **A**daptive **C**on**V**e**X** clustering).

4.2 Related work

The proposed methods are related to some past studies (Zhong and Kwok (2012); Han and Zhang (2015)). We describe the relationships and differences.

For Problem (4.1), we set a new variable $\boldsymbol{v}_m = \boldsymbol{w}_m - \boldsymbol{u}_m$. Then, the minimization problem is converted into the following minimization problem:

$$\min_{\boldsymbol{w}_0,U,V} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{u}_m + \boldsymbol{v}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \|\boldsymbol{v}_m\|_2^2 + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \right\}.$$

This minimization problem is an extension of Problem (2.14): it contains a multi-level structure for the regression coefficient vectors like decomposition or robust approach. This is close to Zhong and Kwok (2012). However, they considered only using the L_1-norm for the fusion of $\boldsymbol{u}_m$ and the squared loss function. The L_1-norm penalty induces feature-level clustering rather than task-level clustering. On the other hand, they also proposed adapting weights for the fused penalty terms. The weights are calculated by using the estimated regression coefficient vectors $\widehat{\boldsymbol{w}}_m$, which may not be better for clustering than calculating the weights using $\widehat{\boldsymbol{u}}_m$, because $\widehat{\boldsymbol{w}}_m$ contains the value of $\widehat{\boldsymbol{v}}_m$.

Moreover, they calculated adaptive weights for all of the combinations. Alternatively, we calculate adaptive weights $\widehat{r}_{m_1,m_2}$ only for $(m_1, m_2) \in \mathcal{E}$.

Han and Zhang (2015) proposed MeTaG (Multi-Level Task Grouping) as follows:

$$\min_{\substack{\boldsymbol{w}_{m,h},\\ m=1,\ldots,T,h=1,\ldots,H}} \left\{ \sum_{m=1}^{T} \frac{1}{2n_m} \|\boldsymbol{y}_m - \boldsymbol{X}_m \sum_{h=1}^{H} \boldsymbol{w}_{m,h}\|_2^2 + \sum_{h=1}^{H} \lambda_h \sum_{m_1>m_2} \|\boldsymbol{w}_{m_1,h} - \boldsymbol{w}_{m_2,h}\|_2 \right\},$$

where $\boldsymbol{w}_{m,h} \in \mathbb{R}^p$ is a parameter vector for m-th task and h-th level, H is a total number of the level. In this minimization problem, the regression coefficient vector $\boldsymbol{w}_m$ is represented by the sum of the h-th level parameter vectors as $\boldsymbol{w}_m = \sum_{h=1}^{H} \boldsymbol{w}_{m,h}$. Furthermore, each h-th level parameter is clustered by the second term. Because the aim of this minimization problem is not to improve the estimation accuracy for regression coefficient vectors and clustering but to capture complex multi-level structures, the proposed methods differ from this method in terms of their aim.

4.3 Estimation algorithm

In the proposed method, we compute the estimates of the parameters by the block coordinate descent algorithm (BCD). The BCD is performed by alternately computing the estimates: $\boldsymbol{u}_m$ is computed given $\boldsymbol{w}_m$, while $\boldsymbol{w}_m$ is done given $\boldsymbol{u}_m$.

We consider the two minimization problems:

$$U^{(t+1)} = \arg\min_{U} \left\{ \frac{\lambda_1}{2} \sum_{m=1}^{T} \|\boldsymbol{w}_m^{(t)} - \boldsymbol{u}_m\|_2^2 + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \right\},$$

$$(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)\top})^\top = \arg\min_{w_{m0},\boldsymbol{w}_m} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \|\boldsymbol{w}_m - \boldsymbol{u}_m^{(t+1)}\|_2^2 \right\}, \quad m = 1, \ldots, T,$$

For the update of $(w_{m0}, \boldsymbol{w}_m)$, it can be solved in a unified manner by the Newton-Raphson method, which is given as Algorithm 3. For the update of $\boldsymbol{u}_m$, we can compute it by using the algorithm for convex clustering such as Shimmura and Suzuki (2022) and Sun et al. (2021). In this thesis, we adopt Algorithm 1 based on the idea of Shimmura

and Suzuki (2022). As a result, the estimation algorithm for Problem (4.1) is given by Algorithm 2. Here, $\mathrm{STL}(\cdot, \cdot)$ is a function returning an estimated regression coefficient vector by an arbitrary single-task learning method. Because MTLCVX is a convex optimization problem and the BCD monotonically decreases the objective function, Algorithm 2 converges to the global minimum. For the convergence criteria, we used $\max_{m=1,\ldots,T, j=1,\ldots,p}(|w_{mj}^{(t)} - w_{mj}^{(t-1)}|)$. If this value is under 0.01, we stop Algorithm 2.

Algorithm 2 Block coordinate descent algorithm for MTLCVX

Require: $\{\boldsymbol{y}_m, X_m; m = 1, \ldots, T\}, k, \lambda_1, \lambda_2$

  for $m = 1, \ldots, T$ **do**

    $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}} = \mathrm{STL}(y_m, X_m)$

  end for

  calculating R by Eq. (4.2) from k and $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$

  $W^{(0)} = \widehat{W}^{\mathrm{STL}}$

  while until convergence of $W^{(t)}$ **do**

    $U^{(t+1)} = \mathrm{CVX}(W^{(t)}, R, \lambda_2/\lambda_1)$

    for $m = 1 \ldots, T$ **do**

      $(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)\top})^\top = \mathrm{NR}(n_m, X_m, \boldsymbol{y}_m, \boldsymbol{u}_m^{(t+1)}, \lambda_1)$

    end for

  end while

Ensure: $U, W, \boldsymbol{w}_0$

Algorithm 3 Newton-Raphson method for updating w_0 and $\boldsymbol{w}_m$

function NR$(n, X, \boldsymbol{y}, \boldsymbol{u}, \lambda_1)$

Initialize; Let $\boldsymbol{X'} = (\mathbf{1}, X), \boldsymbol{w'} = (w_0, \boldsymbol{w}^\top)^\top, \boldsymbol{u'} = (0, \boldsymbol{u}^\top)^\top$, $\Lambda = \mathrm{diag}(0, \lambda_1, \ldots, \lambda_1)$,

$\boldsymbol{\mu} = \left(\frac{\partial b(\eta_1)}{\partial \eta_1}, \ldots, \frac{\partial b(\eta_n)}{\partial \eta_n}\right)^\top, E = \mathrm{diag}\left(\frac{\partial^2 b(\eta_1)}{\partial \eta_1^2}, \ldots, \frac{\partial^2 b(\eta_n)}{\partial \eta_n^2}\right)$.

while until convergence of $\boldsymbol{w'}$ **do**

$$\boldsymbol{w}'^{(t+1)} = \boldsymbol{w}'^{(t)} + \left(\frac{X'^\top E^{(t)} X'}{na(\phi)} + \Lambda\right)^{-1} \left\{\frac{X'^\top (\boldsymbol{y} - \boldsymbol{\mu}^{(t)})}{na(\phi)} - \Lambda(\boldsymbol{w}'^{(t)} - \boldsymbol{u}')\right\}$$

end while

Output: $(w_0, \boldsymbol{w}^\top)^\top = \boldsymbol{w}'$

end function

4.4 Simulation studies

In this section, we report simulation studies in the linear regression setting. We generated data by the true model:

$$\boldsymbol{y}_m = X_m \boldsymbol{w}_m^* + \boldsymbol{\epsilon}_m, \quad m = 1, \ldots, T,$$

where $\boldsymbol{\epsilon}_m$ is an error term whose each component is distributed as $N(0, \sigma^2)$ independently, $\boldsymbol{w}_m^*$ is a true regression coefficient vector for m-th task. For this true model, these T tasks consist of C true clusters. The design matrix X_m was generated from $N_p(\mathbf{0}, \Sigma)$ for each task independently, where $(\Sigma)_{ij} = \phi^{|i-j|}$.

The true regression coefficient vector $\boldsymbol{w}_m^*$ was generated as follows. First, each explanatory variable $\{j = 1, \ldots, p\}$ was randomly assigned to the c-th clusters $\{c = 1, \ldots, C\}$ with the same probability. Then, we generated a true centroid parameter for c-th cluster $\boldsymbol{u}_c^* = (u_{c1}^*, \ldots, u_{cp}^*)^\top$ by

$$u_{cj}^* \begin{cases} \sim N(0, \sigma_u^2) & \text{if } j\text{-th variable is assigned to } c\text{-th cluster,} \\ = 0 & \text{otherwise,} \end{cases} \qquad j = 1, \ldots, p.$$

In addition, we generated a true task-specific parameter for m-th task that belongs to c-th cluster $\boldsymbol{v}_m^{(c)*} = (v_{m1}^{(c)*}, \ldots, v_{mp}^{(c)*})^\top$ by

$$v_{mj}^{(c)*} \begin{cases} \sim N(0, \sigma_v^2) & \text{if } j\text{-th variable is assigned to } c\text{-th cluster,} \\ = 0 & \text{otherwise,} \end{cases} \qquad j = 1, \ldots, p.$$

Finally, we set to $\boldsymbol{w}_m^* = \boldsymbol{u}_c^* + \boldsymbol{v}_m^{(c)*}$. In this way, regression coefficient vectors belonging to different clusters have different non-zero variables. A similar way of generating regression coefficient vectors was also used in Zhou and Zhao (2016).

For the true model, we fixed settings as $n_m = 230$, $p = 100$, $T = 100$, $\sigma^2 = 5$, and $\sigma_u^2 = 100$. 230 samples in each task were split into 30 samples for the train, 100 samples for the validation, and left samples for the test. We considered several settings: $\phi = \{0, 0.2, 0.5\}$, $\sigma_v^2 = \{1, 2, 3, 4, 5\}$, and $C = \{3, 5, 10\}$. Here, when $C = 5$ and 10, the number of tasks in each cluster is uniformly set by T/C. When $C = 3$, that is set as $60, 30$, and 10 tasks, respectively.

To evaluate the effectiveness of our proposed methods, we compared them with the single-task learned lasso (STLL) and the multi-task learning via network lasso (MTLNL). STLL is conducted independently by estimating each task using the lasso. MTLNL is Problem (2.14) for $(l = 1, q = 2)$, which is estimated by the ADMM algorithm of Hallac et al. (2015). The weights r_{m_1,m_2} for both MTLNL and MTLCVX were calculated by Eq. (4.2). In this case, k was set to five. The estimation of both STLL and $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ in Eq. (4.2) were performed by the lasso in R package "glmnet". The tuning parameter ν included in Algorithm 1 and ADMM to estimate MTLNL were set to one. The regularization parameters except for STLL were determined by the validation data. For the evaluation, we calculated the NMSE (normalized mean squared error)

and RMSE (root mean squared error) as follows:

$$\mathrm{NMSE} = \frac{1}{T}\sum_{m=1}^{T}\frac{\|\boldsymbol{y}_m^* - X_m\widehat{\boldsymbol{w}}_m\|_2^2}{\mathrm{Var}(\boldsymbol{y}_{\boldsymbol{m}}^*)},$$

$$\mathrm{RMSE} = \frac{1}{T}\sum_{m=1}^{T}\|\boldsymbol{w}_m^* - \widehat{\boldsymbol{w}}_m\|_2.$$

These values evaluate the accuracy of the prediction and estimated regression coefficient vectors, respectively. They were computed 100 times. The mean and standard deviation were obtained in each setting.

Tables 4.1, 4.2, and 4.3 show the results of the simulation studies for $C = 10$, $C = 5$, and $C = 3$, respectively. Since STLL is independent of the value of σ_v^2, we show the results for STLL only when $\sigma_v^2 = 1$. Note that, according to decreasing the value of C, the number of the true non-zero variables in each task is increased, because variables are nonzero only in the cluster to which they are assigned. Then, the results of STLL in Tables 4.2 and 4.3 considerably deteriorate. This also indicates that the weights r_{m_1,m_2} contain more noise at $C = 3, 5$ than at $C = 10$. Thus, the results of Tables 4.2 and 4.3 are worse than Table 4.1 on the whole.

In a comparison among the methods, MTLACVX shows superior accuracy in almost all situations for both NMSE and RMSE. The differences between MTLACVX and MTLCVX or MTLNL are much larger than those between MTLCVX and MTLNL. Thus, in the context of convex clustering, it means that the adaptive weights are important for improving estimation accuracy. On the other hand, for the comparison of MTLNL and MTLCVX, MTLCVX shows better performance than MTLNL on the whole. In particular, when $C = 5$, MTLCVX is superior to MTLNL in all settings except for NMSE in $\phi = 0$ and $\sigma_v^2 = 5$. When $C = 10$, again, MTLCVX is superior to MTLNL in many settings. MTLNL shows better results than MTLCVX for two settings only when $\phi = 0$. It probably relates the estimation accuracy of $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ to construct wetghts r_{m_1,m_2} by Eq. (4.2). For STLL, RMSE drastically deteriorates by increasing the value of ϕ from 0 to 0.2. This also indicates that the noise in weights r_{m_1,m_2} also

increased from $\phi = 0$ to $\phi = 0.2$. Hence, there is not much difference between MTLNL and MTLCVX for $\phi = 0$, because there was less noise in the weights. However, MTLCVX would be superior to MTLNL as the noise in the weights increased. On the whole, these results suggest that MTLCVX is more robust to the noise in the weights r_{m_1,m_2} than MTLNL. The results in Table 4.3 are obtained in the setting where the number of tasks in each cluster is quite unbalanced. For these settings, MTLCVX is superior to MTLNL in many settings when $\phi = 0.2$ or 0.5, while the opposite is true when $\phi = 0$. Because the RMSE of STLL has little difference among each ϕ, the noise in weights is probably the almost same. Therefore, the reasons why MTLCVX is superior at a larger value of ϕ would differ from those in Tables 4.1 and 4.2. On the other hand, the reason that MTLNL is superior to MTLCVX for $\phi = 0$ may relate to the estimation of cluster centers. MTLCVX assumes the existence of cluster centers and estimates them, while MTLNL does not need to perform estimating the cluster centers. In this setting, the minimum number of tasks in a cluster is ten, which may be insufficient to estimate cluster centers.

Table 4.1: Mean and standard deviation of NMSE and RMSE for $C = 10$.

		$\phi = 0$		$\phi = 0.2$		$\phi = 0.5$	
σ_v^2		NMSE	RMSE	NMSE	RMSE	NMSE	RMSE
–	STLL	0.200 (0.025)	1.494 (0.161)	0.198 (0.034)	4.062 (0.284)	0.178 (0.033)	4.070(0.315)
1	MTLNL	0.059 (0.028)	0.646 (0.146)	0.053 (0.032)	0.614 (0.148)	0.049 (0.029)	0.664 (0.176)
	MTLCVX	0.055 (0.038)	0.609 (0.190)	0.048 (0.023)	0.596 (0.155)	0.039 (0.018)	0.574 (0.147)
	MTLACVX	**0.044** (0.019)	**0.565** (0.158)	**0.043** (0.029)	**0.559** (0.152)	**0.038** (0.023)	**0.565** (0.156)
2	MTLNL	0.075 (0.029)	0.741 (0.127)	0.068 (0.024)	0.755 (0.163)	0.063 (0.030)	0.762 (0.137)
	MTLCVX	0.063 (0.026)	**0.689** (0.148)	0.058 (0.023)	0.667 (0.151)	0.052 (0.021)	0.691 (0.138)
	MTLACVX	**0.060** (0.024)	0.696 (0.185)	**0.055** (0.020)	**0.637** (0.125)	**0.048** (0.020)	**0.666** (0.124)
3	MTLNL	0.083 (0.049)	0.789 (0.160)	0.080 (0.023)	0.815 (0.130)	0.076 (0.036)	0.868 (0.141)
	MTLCVX	**0.080** (0.035)	0.775 (0.157)	0.078 (0.038)	0.767 (0.119)	0.066 (0.021)	0.791 (0.126)
	MTLACVX	0.081 (0.035)	**0.771** (0.138)	**0.073** (0.024)	**0.752** (0.124)	**0.065** (0.027)	**0.764** (0.122)
4	NLMTL	0.106 (0.077)	0.906 (0.130)	0.093 (0.026)	0.889 (0.138)	0.079 (0.020)	0.906 (0.090)
	MTLCVX	**0.084** (0.027)	0.818 (0.111)	0.090 (0.033)	0.856 (0.122)	0.076 (0.024)	0.861 (0.123)
	MTLACVX	0.085 (0.025)	**0.815** (0.126)	**0.084** (0.024)	**0.831** (0.129)	**0.074** (0.024)	**0.841** (0.112)
5	MTLNL	0.105 (0.029)	0.921 (0.099)	0.102 (0.025)	0.939 (0.111)	0.096 (0.043)	0.998 (0.118)
	MTLCVX	0.113 (0.099)	0.906 (0.129)	0.099 (0.030)	0.898 (0.124)	0.088 (0.025)	0.938 (0.109)
	MTLACVX	**0.099** (0.032)	**0.888** (0.120)	**0.094** (0.025)	**0.894** (0.130)	**0.087** (0.027)	**0.931** (0.114)

Table 4.2: Mean and standard deviation of NMSE and RMSE for $C = 5$.

		$\phi = 0$		$\phi = 0.2$		$\phi = 0.5$	
σ_v^2		NMSE	RMSE	NMSE	RMSE	NMSE	RMSE
–	STLL	0.567 (0.046)	3.423 (0.284)	0.564 (0.054)	5.241 (0.055)	0.493 (0.054)	5.395(0.359)
1	MTLNL	0.127 (0.059)	1.543 (0.394)	0.117 (0.055)	1.475 (0.388)	0.080 (0.048)	1.337 (0.396)
	MTLCVX	0.131 (0.055)	1.559 (0.384)	0.113 (0.055)	1.459 (0.379)	0.074 (0.041)	1.291 (0.393)
	MTLACVX	**0.105** (0.053)	**1.361** (0.430)	**0.112** (0.056)	**1.415** (0.403)	**0.070** (0.039)	**1.225** (0.372)
2	MTLNL	0.145 (0.054)	1.625 (0.350)	0.139 (0.055)	1.627 (0.373)	0.086 (0.043)	1.435 (0.377)
	MTLCVX	0.142 (0.058)	1.606 (0.339)	0.131 (0.045)	1.582 (0.318)	0.078 (0.043)	1.296 (0.352)
	MTLACVX	**0.132** (0.062)	**1.557** (0.423)	**0.112** (0.056)	**1.470** (0.380)	**0.078** (0.041)	**1.293** (0.336)
3	MTLNL	0.151 (0.056)	1.689 (0.339)	0.146 (0.051)	1.710 (0.334)	0.102 (0.035)	1.545 (0.308)
	MTLCVX	0.159 (0.054)	1.730 (0.339)	0.134 (0.061)	1.582 (0.384)	**0.090** (0.038)	**1.442** (0.297)
	MTLACVX	**0.132** (0.054)	**1.532** (0.369)	**0.119** (0.049)	**1.495** (0.316)	0.094 (0.045)	1.444 (0.373)
4	MTLNL	0.162 (0.054)	1.774 (0.339)	0.162 (0.057)	1.801 (0.344)	0.108 (0.040)	1.575 (0.287)
	MTLCVX	0.155 (0.060)	1.716 (0.351)	0.154 (0.056)	1.746 (0.340)	0.099 (0.041)	1.533 (0.340)
	MTLACVX	**0.145** (0.059)	**1.667** (0.397)	**0.120** (0.044)	**1.515** (0.309)	**0.094** (0.044)	**1.460** (0.364)
5	MTLNL	0.179 (0.061)	1.853 (0.330)	0.169 (0.051)	1.850 (0.303)	0.117 (0.040)	1.669 (0.298)
	MTLCVX	0.163 (0.057)	1.757 (0.315)	0.157 (0.051)	1.773 (0.311)	0.119 (0.046)	1.674 (0.323)
	MTLACVX	**0.146** (0.041)	**1.706** (0.274)	**0.130** (0.042)	**1.612** (0.276)	**0.093** (0.036)	**1.481** (0.288)

Table 4.3: Mean and standard deviation of NMSE and RMSE for $C = 3$.

		$\phi = 0$		$\phi = 0.2$		$\phi = 0.5$	
σ_v^2		NMSE	RMSE	NMSE	RMSE	NMSE	RMSE
–	STLL	0.797 (0.047)	6.669 (0.461)	0.790 (0.047)	6.649 (0.456)	0.705 (0.055)	6.773 (0.452)
1	MTLNL	0.208 (0.064)	2.544 (0.458)	0.201 (0.058)	2.577 (0.388)	0.140 (0.052)	2.386 (0.469)
	MTLCVX	0.209 (0.076)	2.554 (0.476)	0.211 (0.081)	2.556 (0.470)	0.117 (0.067)	2.092 (0.572)
	MTLACVX	**0.198** (0.072)	**2.454** (0.468)	**0.171** (0.069)	**2.327** (0.446)	**0.104** (0.058)	**1.935** (0.545)
2	MTLNL	0.206 (0.062)	2.530 (0.392)	0.194 (0.058)	2.502 (0.404)	0.143 (0.066)	2.322 (0.532)
	MTLCVX	0.221 (0.067)	2.641 (0.410)	0.204 (0.060)	2.612 (0.432)	0.134 (0.067)	2.278 (0.631)
	MTLACVX	**0.193** (0.072)	**2.446** (0.472)	**0.180** (0.065)	**2.393** (0.477)	**0.109** (0.061)	**2.024** (0.590)
3	MTLNL	0.226 (0.059)	2.683 (0.380)	0.231 (0.061)	2.736 (0.372)	0.150 (0.056)	2.407 (0.454)
	MTLCVX	0.227 (0.067)	2.662 (0.404)	0.220 (0.066)	2.690 (0.399)	0.127 (0.057)	2.261 (0.485)
	MTLACVX	**0.205** (0.091)	**2.499** (0.460)	**0.193** (0.076)	**2.490** (0.521)	**0.125** (0.063)	**2.167** (0.563)
4	MTLNL	0.233 (0.064)	2.751 (0.418)	0.223 (0.064)	2.720 (0.395)	0.149 (0.054)	2.479 (0.484)
	MTLCVX	0.237 (0.068)	2.775 (0.397)	0.222 (0.067)	2.714 (0.402)	0.145 (0.059)	2.416 (0.496)
	MTLACVX	**0.213** (0.064)	**2.618** (0.435)	**0.199** (0.060)	**2.490** (0.521)	**0.128** (0.053)	**2.246** (0.415)
5	MTLNL	0.240 (0.057)	2.852 (0.376)	0.238 (0.057)	2.870 (0.397)	0.157 (0.056)	2.557 (0.431)
	MTLCVX	0.256 (0.070)	2.865 (0.377)	0.234 (0.060)	2.789 (0.389)	0.158 (0.058)	2.516 (0.463)
	MTLACVX	**0.214** (0.072)	**2.620** (0.450)	**0.189** (0.064)	**2.541** (0.433)	**0.126** (0.051)	**2.226** (0.453)

4.5 Application to real datasets

In this section, we applied our proposed methods to two datasets with continuous and binary responses. The first is the school data (Bakker and Heskes, 2003), which has been often used as the research of an MTL. This dataset consists of examination scores of 15,362 students, four school-specific attributes, and three student-specific attributes from 139 secondary schools in London from 1985 to 1987. The dataset was obtained by "MALSAR" package in MATLAB. In the package, categorical attributes were replaced with binary attributes. Then, we used 28-dimensional explanatory variables and the examination scores as a response. Each school is considered as a task. The second is the landmine data (Xue et al., 2007), which consists of nine-dimensional features and the corresponding binary labels for 29 tasks. One task corresponds to one landmine field

where data were collected: 1–15 tasks correspond to regions that are relatively highly foliated and 16–29 tasks correspond to regions that are bare earth or desert. Therefore, there may be two clusters depending on the ground surface conditions. The responses represent landmines or clutter. The features are four moment-based features, three correlation-based features, one energy ratio feature, and one spatial variance feature, which are extracted from radar images. Though there are 14,820 samples in total, this dataset is quite unbalanced: positive samples are few, while negative ones are many. To perform our proposed method, down-sampling was done by reducing negative samples to equal the number of positive samples. In the results, we used 1,808 samples in total.

We compared our proposed methods MTLCVX, MTLACVX with MTLNL, STLL, and single-task learned ridge (STLR) in prediction accuracy. Here, STLR is the ridge estimation performed by R package "glmnet" for each task, independently. Note that, to stabilize estimation in the logistic regression of MTLNL, MTLCVX, and MTLACVX, we penalized the intercept w_{m0} by the ridge. Its regularization parameter was set to 0.1. This penalty has the effect of keeping the intercept constant finite stable value in the situation that the intercept tends to go to infinity. We randomly split the data into $V\%$ of the data for the train, $(80 - V)\%$ for the test, and 20% for the validation. We conducted three settings $V = \{50, 60, 70\}$. For the evaluation, we used NMSE for analyzing the school data, while we used AUC for analyzing the landmine data. The mean and standard deviation of evaluation values were computed from 100 repetitions. The tuning parameter k in Eq. (4.2) was set to five for all MTL methods and $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ were estimated by the lasso by the package "glmnet" in R.

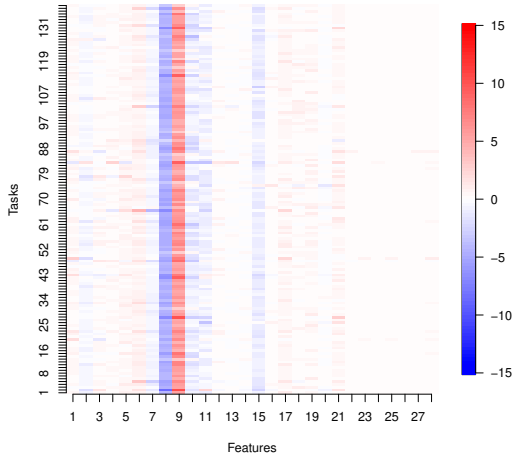
In addition, we considered the difference in patterns of estimated parameters among MTLNL, MTLCVX, and MTLACVX for both school data and down-sampled landmine data. For the comparison, we split the dataset into 70% samples for the train and 30% samples for the validation. The regularization parameters were determined by these split data. Then, the models were refitted by using all samples and the determined reg-

ularization parameters. Note that the randomness in data splitting and down-sampling were fixed in all settings for stable comparison.

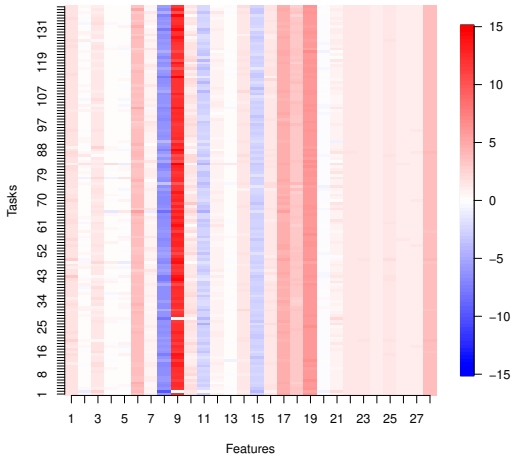
Figures 4.1 and 4.2 show results of estimated parameters in MTLNL and MTLCVX for the school data and the landmine data, respectively. In these figures, the vertical axis is the index of tasks m and the horizontal axis is the index of features j in $\widehat{w}_{mj}$ or $\widehat{u}_{mj}$. The color represents the value of the estimated parameter in its coordinates. Here, in Figures 4.2(a) and 4.2(b), the first column corresponds to the intercepts $\widehat{w}_{m0}$ and others correspond to $\widehat{\boldsymbol{w}}_m$. In Figure 4.1, MTLNL in (a) and MTLCVX in (b) show similar homogeneous patterns in regression coefficients and MTLCVX has more uniform patterns than MTLNL. On the other hand, those of $\widehat{\boldsymbol{w}}_m$ in (b) and $\widehat{\boldsymbol{u}}_m$ in (c) for MTLCVX are almost same, which indicates that $\widehat{\boldsymbol{w}}_m$ are greatly affected by $\widehat{\boldsymbol{u}}_m$. These differences between MTLCVX and MTLNL are probably caused by the fact that only MTLCVX assumes the existence of cluster centers, which induces homogeneity of $\widehat{\boldsymbol{u}}_m$ among all tasks. In Figure 4.2, MTLNL in (a) and MTLCVX in (b) show quite different patterns of regression coefficients. The patterns of MTLNL are rather homogenous. In contrast, those of MTLCVX show the existence of clusters in regression coefficients. Furthermore, we can see clear two cluster patterns from the value of $\widehat{\boldsymbol{u}}_m$ in (c). These patterns except for 20-th task are consistent with the fact that 1–15 tasks and 16–29 tasks are collected from different types of ground surface conditions. Therefore, it would be considered that MTLCVX improves the problem of MTLNL about shrinkage between different clusters. In addition, the pattern of 20-th task $\widehat{\boldsymbol{u}}_{20}$ shows similar result to those of 1–15 tasks. This may suggest that the 20-th task is unique among 16–29 tasks collected from the regions that are bare earth or desert and represents similar characteristics to 1–15 tasks. On the whole, it is interesting that MTLCVX shows a clearer cluster structure than MTLNL, whether there is a single cluster as in the case of the school data, or multiple clusters as in the case of the landmine data.

The results of estimated parameters for MTLACVX are shown in Figures 4.3 and

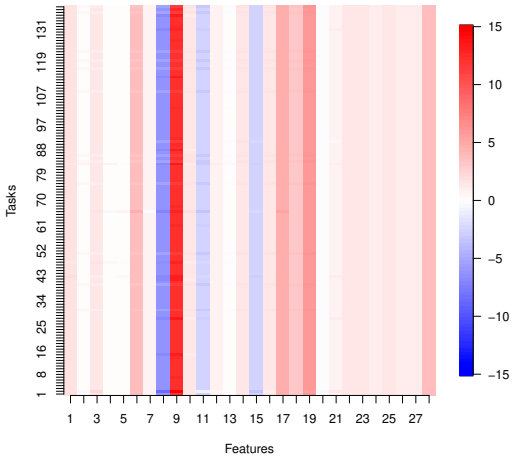
4.4, which are similar to and more clear patterns than those of MTLCVX. These results would demonstrate that MTLACVX improves MTLCVX in terms of clustering.



(a) Estimated value of $\widehat{\boldsymbol{w}}_m$ in MTLNL

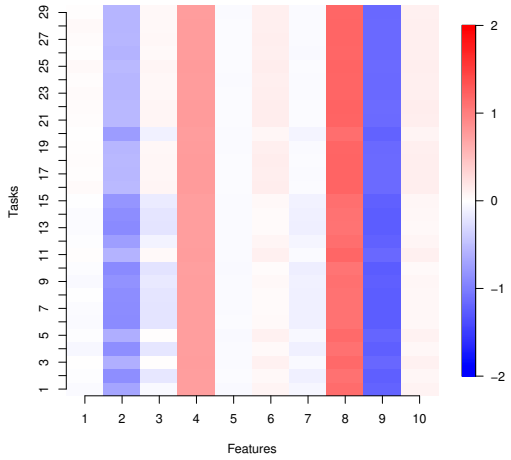


(b) Estimated value of $\widehat{\boldsymbol{w}}_m$ in MTLCVX

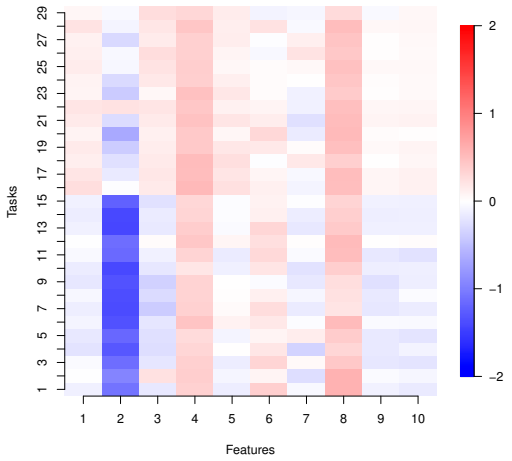


(c) Estimated value of $\widehat{\boldsymbol{u}}_m$ in MTLCVX

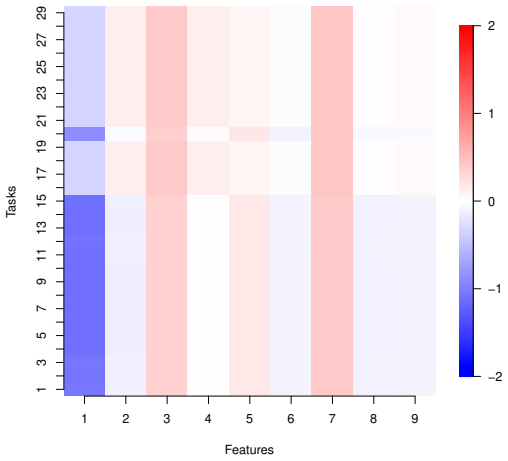
Figure 4.1: The estimated value of parameters in MTLNL and MTLCVX for the school data.



(a) Estimated value of $\widehat{w}_{m0}$ and $\widehat{\boldsymbol{w}}_m$ in MTLNL

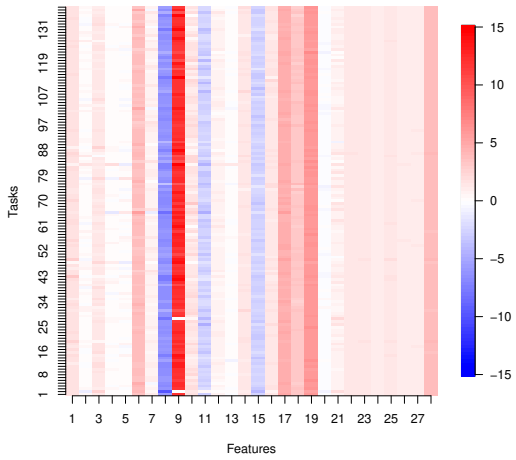


(b) Estimated value of $\widehat{w}_{m0}$ and $\widehat{\boldsymbol{w}}_m$ in MTLCVX

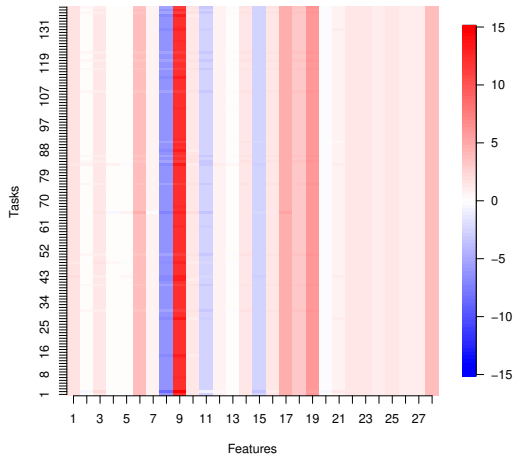


(c) Estimated value of $\widehat{\boldsymbol{u}}_m$ in MTLCVX

Figure 4.2: The estimated value of parameters in MTLNL and MTLCVX for the landmine data.

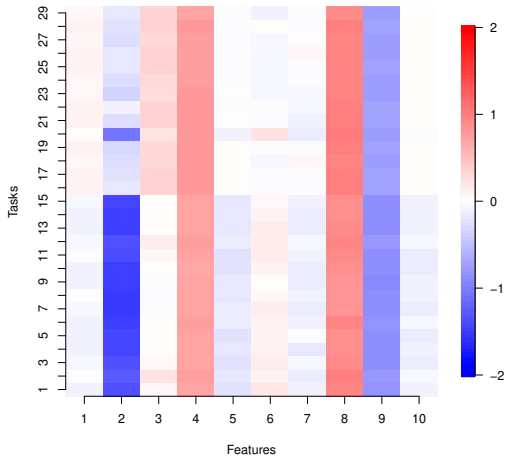


(a) Estimated value of $\widehat{\boldsymbol{w}}_m$ in MTLACVX

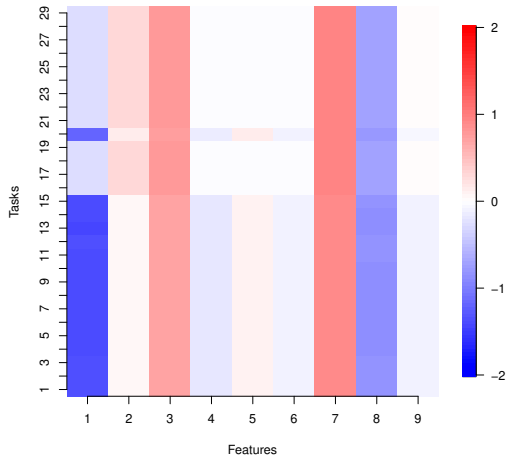


(b) Estimated value of $\widehat{\boldsymbol{u}}_m$ in MTLACVX

Figure 4.3: The estimated value of parameters in MTLACVX for the school data.



(a) Estimated value of $\widehat{\boldsymbol{w}}_m$ in MTLACVX



(b) Estimated value of $\widehat{\boldsymbol{u}}_m$ in MTLACVX

Figure 4.4: The estimated value of parameters in MTLACVX for the landmine data.

Table 4.4 shows the results of the comparison of NMSE on the school data for each setting. First, all MTL methods are superior to single-task learning approaches. In a comparison among MTL methods, each method shows a better result for each setting.

However, because all settings have outstanding standard deviations for $V = 70$, this result is probably not trustworthy. While it is pointed out by Evgeniou et al. (2005) and shown in Figure 4.1 that the school data do not have multiple clusters and are rather homogenous, MTLCVX may foster excessive homogeneity among all tasks compared to MTLNL in cases where only a single cluster is present.

Table 4.5 shows the results of the comparison of AUC on the landmine data for each setting. In the data, MTLACVX and MTLCVX are superior to STL methods and MTLNL for all settings. MTLACVX also has the same or better performance than MTLCVX. Unlike the school data, the landmine data are considered to have clearly two clusters as shown in Figure 4.2. This would be the reason that MTLCVX and MTLACVX in the landmine data provide higher accuracy compared to those in the school data.

Table 4.4: Mean and standard deviation of NMSE for 100 repetitions in the school data.

	50%	60%	70%
STLL	4.044 (0.181)	4.293 (0.234)	5.783 (1.306)
STLR	4.701 (0.226)	5.071 (0.516)	6.533 (1.170)
MTLNL	0.806 (0.025)	**0.847** (0.036)	1.196 (0.517)
MTLCVX	**0.796** (0.025)	0.853 (0.060)	1.241 (0.825)
MTLACVX	0.830 (0.036)	0.863 (0.060)	**1.140** (0.528)

Table 4.5: Mean and standard deviation of AUC for 100 repetitions in the landmine data.

	50%	60%	70%
STLL	0.746 (0.023)	0.748 (0.022)	0.748 (0.020)
STLR	0.749 (0.023)	0.750 (0.023)	0.749 (0.027)
MTLNL	0.754 (0.024)	0.749 (0.024)	0.750 (0.021)
MTLCVX	**0.769** (0.021)	0.759 (0.020)	0.760 (0.023)
MTLACVX	0.768 (0.018)	**0.764** (0.022)	**0.770** (0.023)

4.6 Discussion

In this chapter, we considered reducing the incorrect shrinkage between unrelated tasks while maintaining the problem as convex optimization. To this end, we focused on separating the regression coefficients rather than improving the regularization weights. The simulation studies and application to real datasets showed that introduced centroid parameters can not only improve estimation accuracy but also allow us to interpret cluster structures more clearly than MTLNL. This helps understand relationships among tasks, particularly when estimated regression coefficients show complicated patterns. Additionally, adapting the regularization weights as in the adaptive lasso showed improved estimation accuracy in many situations.

One limitation of the proposed methods is their computational complexity. Specifically, we used the BCD to estimate parameters, which requires one ADMM loop for an update of $U^{(t+1)}$ in each iteration. This is quite inefficient compared to MTLNL, which can be estimated with only one ADMM loop. This problem will be addressed in the next chapter by incorporating the update of $W^{(t+1)}$ into the ADMM loop. In addition, our methods require determining two regularization parameters, making computation much more expensive than methods with a single parameter. These factors contribute

to high computational costs in both the estimation algorithm and parameter tuning.

Another important consideration is the method of weight construction. Our study employed a k-nearest neighbor method based on existing MTL methods and convex clustering literature. While this approach showed effectiveness, there may be more efficient methods for constructing weights regarding both computational complexity and estimation accuracy. Recent work by Zhang et al. (2024) proposed combining minimum spanning trees with the weights calculated by the OLS, demonstrating consistency in estimating latent clusters under asymptotic conditions. However, multi-task learning typically deals with limited samples per task or only some tasks approaching asymptotic conditions. Therefore, developing optimal weight construction methods for finite samples remains challenging.

We leave these topics of computational efficiency and weight construction methodology as important future work.

Chapter 5

Multi-task learning with joint estimation of clusters and detection of outlier tasks

Many MTL methods have utilized clustering techniques to treat heterogeneous characteristics within a task set. However, clustering techniques employed for MTL methods are not robust for outlier samples. For example, the k-means method is considered as sensitive to the outlier samples due to its property, which is that k-means intends to make the size of clusters equal. Thus, it would be a natural extension to make the clustering-based MTL methods robust for outlier tasks.

In this chapter, we first demonstrate robust convex clustering (Quan and Chen, 2020), which is formulated as convex clustering with additional outlier parameters and L_1-penalty for them. Then, we construct the relationship between the formulation based on L_1-penalty and the Huber-loss function, which is further generalized to the relationship between wide penalty functions and M-estimators with multivariate robust loss functions. Based on them, we propose a robust MTL method that simultaneously detects outlier tasks and performs clustering of tasks.

5.1 Robust convex clustering

Quan and Chen (2020) pointed out that convex clustering is sensitive to just a few outliers. To address this issue, they proposed robust convex clustering (RCC) as follows:

$$\min_{U,O}\left\{\sum_{i=1}^{n}\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{u}_i-\boldsymbol{o}_i\|_2^2+\lambda_1\sum_{(i_1,i_2)\in\mathcal{E}}r_{i_1,i_2}\|\boldsymbol{u}_{i_1}-\boldsymbol{u}_{i_2}\|_2+\lambda_2\sum_{i=1}^{n}\|\boldsymbol{o}_i\|_1\right\}, \tag{5.1}$$

where $\boldsymbol{u}_i=(u_{i1},\ldots,u_{ip})^\top\in\mathbb{R}^p$ is a vector of centroid parameters for i-th sample, $\boldsymbol{o}_i=(o_{i1},\ldots,o_{ip})^\top\in\mathbb{R}^p$ is a vector of outlier parameters for i-th sample, $U=(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n)^\top$ and $O=(\boldsymbol{o}_1,\ldots,\boldsymbol{o}_n)^\top$ are $n\times p$ matrices, respectively. The third term selects the outlier parameters by shrinking each element of $\boldsymbol{o}_i$ toward exactly zero. If o_{ij} is estimated to be a non-zero value, the j-th feature of the i-th sample is considered an outlier.

From the relationship between the loss function of least square and L_1 penalty for the outlier parameters (Antoniadis (2007); Gannaz (2007)), the minimization problem (5.1) is equivalent to the following minimization problem:

$$\min_{U}\left\{\sum_{i=1}^{n}\sum_{j=1}^{p}h_{\lambda_2}(x_{ij}-u_{ij})+\lambda_1\sum_{(i_1,i_2)\in\mathcal{E}}r_{i_1,i_2}\|\boldsymbol{u}_{i_1}-\boldsymbol{u}_{i_2}\|_2\right\}, \tag{5.2}$$

where $h_\lambda(\cdot)$ is the Huber's loss function defined as

$$h_{\lambda_2}(z)=\begin{cases}\frac{1}{2}z^2 & |z|\le\lambda,\\ \lambda_2|z|-\frac{\lambda^2}{2} & |z|\ge\lambda.\end{cases}$$

This would indicate that RCC is a robust version of convex clustering derived from replacing the loss function with the component-wise robust loss function.

5.2 Non-convex extensions of robust convex clustering

Although Quan and Chen (2020) only considered L_1 penalty to select outlier parameters, it is possible to consider other types of shrinkage penalties, such as non-convex

penalties and group penalties. For example, if group lasso (Yuan and Lin, 2006) is employed for the third term in (5.1), we can detect the sample-wise outliers. Since our purpose is to detect task-wise outliers, we first consider the generalized robust clustering problem to detect sample-wise outliers as follows:

$$\min_{U,O} \left\{ \sum_{i=1}^{n} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{u}_i - \boldsymbol{o}_i\|_2^2 + \lambda_1 \sum_{(i_1,i_2)\in\mathcal{E}} r_{i_1,i_2} \|\boldsymbol{u}_{i_1} - \boldsymbol{u}_{i_2}\|_2 + \sum_{i=1}^{n} P(\boldsymbol{o}_i; \lambda_2, \gamma) \right\}, \quad (5.3)$$

where $P(\cdot; \lambda, \gamma)$ is a penalty function that induces group sparsity and γ is a tuning parameter that adapts the shape of the penalty function. By estimating $\boldsymbol{o}_i$ as a zero vector through group penalties, this problem aims to detect sample-wise outliers. For i-th sample $\boldsymbol{x}_i$, even if the value of one feature x_{ij} has extensive value compared with its cluster center $\widehat{u}_{ij}$, $\widehat{\boldsymbol{o}}_i$ would not be non-zero vector. Only when the L_2-distance $\|\boldsymbol{x}_i - \widehat{\boldsymbol{u}}_i\|_2$ has the extensive value, $\widehat{\boldsymbol{o}}_i$ is estimated to be non-zero vector and $\boldsymbol{x}_i$ is interpreted as an outlier sample.

If a non-convex penalty such as group SCAD and group MCP (Huang et al., 2012) is employed for the third term, the minimization problem is no longer a convex optimization problem. Therefore, we refer to the minimization problem as **R**obust **R**egularized **C**lustering (RRC).

Algorithm 4 Block coordinate descent algorithm for Problem (5.3)

Require: $X, \lambda_1, \lambda_2, \gamma, O^{(0)}$

while until convergence of $U^{(t)}$ and $O^{(t)}$ **do**

$$U^{(t+1)} = \arg\min_{U} L(U, O^{(t)}) \quad (5.4)$$

$$O^{(t+1)} = \arg\min_{O} L(U^{(t+1)}, O) \quad (5.5)$$

end while

Ensure: U, O

The minimization problem (5.3) is solved by the BCD algorithm shown in Algorithm 4. Here, $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n \times p}$, $L(U, O)$ is the objective function in (5.3). The problem (5.4) can be solved by algorithms for the convex clustering. The problem (5.5) can be updated separately in terms of $\boldsymbol{o}_i$, because each $\boldsymbol{o}_i$ only depends on $\boldsymbol{x}_i$ and $\boldsymbol{u}_i^{(t+1)}$. Thus, the update is expressed as

$$\boldsymbol{o}_i^{(t+1)} = \arg\min_{\boldsymbol{o}_i} \left\{ \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{u}_i^{(t+1)} - \boldsymbol{o}_i\|_2^2 + P(\boldsymbol{o}_i; \lambda_2, \gamma) \right\}, \quad i = 1, \ldots, n. \tag{5.6}$$

Therefore, the update can be obtained by

$$\boldsymbol{o}_i^{(t+1)} = \boldsymbol{\Theta}(\boldsymbol{x}_i - \boldsymbol{u}_i^{(t+1)}; \lambda_2, \gamma), \quad i = 1, \ldots, n,$$

where $\boldsymbol{\Theta}(\cdot; \lambda, \gamma)$ is a group-thresholding function defined for the corresponding penaly function $P(\cdot; \lambda, \gamma)$. For example, $\boldsymbol{\Theta}(\cdot; \lambda, \gamma)$ for group lasso is given by

$$\boldsymbol{\Theta}^{\text{glasso}}(\boldsymbol{z}; \lambda, \gamma) = S(\boldsymbol{z}; \lambda),$$

where $S(\cdot; \lambda)$ is a group soft-thresholding function defined as

$$S(\boldsymbol{z}; \lambda) = \max\left(0, 1 - \frac{\lambda}{\|\boldsymbol{z}\|_2}\right) \boldsymbol{z}.$$

The solution of (5.3) obtained by Algorithm 4 is related to M-estimators, which is similar to the connection between the minimization problems (5.1) and (5.2). Let A_r be a $|\mathcal{E}| \times n$ matrix whose each row is $r_{i_1,i_2} \boldsymbol{a}_{(i_1,i_2)}^\top$ and we set

$$D_r = A_r \otimes I_p,$$

where $\otimes$ denotes the Kronecker product, I_p is a $p \times p$ identity matrix, and $\boldsymbol{a}_{(i_1,i_2)}$ is defined as (3.6). We also define the mixed $(2,1)$-norm (Lounici et al., 2011) for a $|\mathcal{E}|p$-dimensional vector $\boldsymbol{z}$ as

$$\|\boldsymbol{z}\|_{2,1} = \sum_{k=1}^{|\mathcal{E}|} \left(\sum_{j=(k-1)p+1}^{kp} z_j^2 \right)^{1/2}.$$

Using these definitions, the second term in (5.3) can be written as

$$\sum_{(i_1,i_2)\in\mathcal{E}} r_{i_1,i_2}\|\boldsymbol{u}_{i_1}-\boldsymbol{u}_{i_2}\|_2 = \|D_r\mathrm{vec}(U)\|_{2,1}.$$

Based on the above definitions, we summarize the relationship between the solution of the RRC and M-estimator in the following proposition.

Proposition 1. *Suppose that $\widehat{U}$ is a convergence point in Algorithm 4 and $\boldsymbol{\psi}(\boldsymbol{o};\lambda,\gamma) = \boldsymbol{o} - \boldsymbol{\Theta}(\boldsymbol{o};\lambda,\gamma)$. Then, the $\widehat{U}$ satisfies*

$$-\Psi(X-\widehat{U};\lambda_2,\gamma) + \lambda_1\partial_{\mathrm{vec}(U)}(\|D_r\mathrm{vec}(U)\|_{2,1})|_{U=\widehat{U}} \ni \mathbf{0}, \tag{5.7}$$

where $\partial_{\mathrm{vec}(U)}$ is the subdifferential with respect to $\mathrm{vec}(U)$*, and $\Psi(X-\widehat{U};\lambda_2,\gamma)$ is an np-dimensional vector defined as*

$$\Psi(X-\widehat{U};\lambda_2,\gamma) = \begin{pmatrix} \boldsymbol{\psi}(\boldsymbol{x}_1-\widehat{\boldsymbol{u}}_1;\lambda_2,\gamma) \\ \boldsymbol{\psi}(\boldsymbol{x}_2-\widehat{\boldsymbol{u}}_2;\lambda_2,\gamma) \\ \vdots \\ \boldsymbol{\psi}(\boldsymbol{x}_n-\widehat{\boldsymbol{u}}_n;\lambda_2,\gamma) \end{pmatrix}.$$

The proof of the proposition is given by Section 5.8.2. This proposition gives the relationship between the $\widehat{U}$ calculated by Algorithm 4 and the following minimization problem:

$$\min_U \left\{ \sum_{i=1}^n \rho_{\lambda_2,\gamma}(\boldsymbol{x}_i-\boldsymbol{u}_i) + \lambda_1 \sum_{(i_1,i_2)\in\mathcal{E}} r_{i_1,i_2}\|\boldsymbol{u}_{i_1}-\boldsymbol{u}_{i_2}\|_2 \right\}, \tag{5.8}$$

where $\rho_{\lambda,\gamma}(\cdot)$ is a multivariate loss function that satisfies

$$\frac{\partial}{\partial \boldsymbol{z}}\rho_{\lambda,\gamma}(\boldsymbol{z}) = \boldsymbol{\psi}(\boldsymbol{z};\lambda,\gamma).$$

In general, a subgradient for a function involving a non-convex function term does not satisfy the sum rule. Thus, the optimality condition for Problem (5.8) with non-convex function $\rho_{\lambda,\gamma}(\cdot)$ does not coincide with the inclusion relationship (5.7). However, if we choose a regularization term $P(\boldsymbol{z};\lambda,\gamma)$ so that $\rho_{\lambda,\gamma}(\boldsymbol{z})$ is a weakly convex function,

the sum rule of the subdifferential (Ngai et al. (2000); Corollary 3.9) can be applied to the objective function of Problem (5.8). Here, a function $f(\boldsymbol{z})$ is called C_p-weakly convex function with modules $C_p \geq 0$, if the function $g(\boldsymbol{z}) = f(\boldsymbol{z}) + \frac{C_p}{2}\|\boldsymbol{z}\|_2^2$ is a convex function (e.g. Denevi et al. (2018)). With this weak convexity of $\rho_{\lambda,\gamma}(\cdot)$, the optimality condition regarding the stationary points coincides with the inclusion relationship (5.7), which means any output $\widehat{U}$ is one of the stationary points of the Problem (5.8). The loss functions induced from group SCAD and group MCP are weakly convex, while the skipped mean loss (e.g. Hampel (1985)) induced from group hard thresholding is not weakly convex. The proofs of the weakly convexity of those loss functions are given in Section 5.8.1.

We present four specific multivariate loss functions and their corresponding group-thresholding functions. Some of them are illustrated in Figure 5.1.

Multivariate loss functions and group-thresholding functions

Group SCAD

For $\gamma > 2$, the group SCAD thresholding function and loss function are, respectively, expressed as

$$\rho_{\lambda,\gamma}^{\mathrm{gSCAD}}(\boldsymbol{z}) = \begin{cases} \frac{1}{2}\|\boldsymbol{z}\|_2^2 & \|\boldsymbol{z}\|_2 \leq \lambda, \\ \lambda\|\boldsymbol{z}\|_2 - \frac{\lambda^2}{2} & \lambda \leq \|\boldsymbol{z}\|_2 < 2\lambda, \\ \frac{\gamma\lambda}{\gamma-2}\|\boldsymbol{z}\|_2 - \frac{1}{2(\gamma-2)}\|\boldsymbol{z}\|_2^2 - \frac{\gamma+2}{2(\gamma-2)}\lambda^2 & 2\lambda \leq \|\boldsymbol{z}\|_2 \leq \gamma\lambda, \\ \frac{\gamma+1}{2}\lambda^2 & \gamma\lambda < \|\boldsymbol{z}\|_2, \end{cases}$$

$$\boldsymbol{\Theta}^{\mathrm{gSCAD}}(\boldsymbol{o}; \lambda, \gamma) = \begin{cases} S(\boldsymbol{o}; \lambda), & \|\boldsymbol{o}\|_2 \leq 2\lambda, \\ \frac{\gamma-1}{\gamma-2} S(\boldsymbol{o}, \frac{\gamma\lambda}{\gamma-1}), & 2\lambda < \|\boldsymbol{o}\|_2 \leq \gamma\lambda, \\ \boldsymbol{o}, & \|\boldsymbol{o}\|_2 > \gamma\lambda. \end{cases}$$

Group MCP

For $\gamma > 1$, the group MCP thresholding function and loss function are, respectively, expressed as

$$\rho^{\mathrm{gMCP}}_{\lambda,\gamma}(\boldsymbol{z}) = \begin{cases} \frac{1}{2}\|\boldsymbol{z}\|_2^2 & \|\boldsymbol{z}\|_2 \leq \lambda, \\ \frac{\gamma\lambda}{\gamma-1}\|\boldsymbol{z}\|_2 - \frac{1}{2(\gamma-1)}\|\boldsymbol{z}\|_2^2 - \frac{\gamma\lambda^2}{2(\gamma-1)} & \lambda \leq \|\boldsymbol{z}\|_2 \leq \gamma\lambda, \\ \frac{\gamma\lambda^2}{2} & \gamma\lambda < \|\boldsymbol{z}\|_2, \end{cases}$$

$$\boldsymbol{\Theta}^{\mathrm{gMCP}}(\boldsymbol{o}; \lambda, \gamma) = \begin{cases} \frac{\gamma}{\gamma-1} S(\boldsymbol{o}, \lambda), & \|\boldsymbol{o}\|_2 \leq \gamma\lambda, \\ \boldsymbol{o}, & \|\boldsymbol{o}\|_2 > \gamma\lambda. \end{cases}$$

Multivariate skipped mean loss

We define the multivariate version of the skipped mean loss function as

$$\rho^{\mathrm{MS}}_{\lambda}(\boldsymbol{z}) = \begin{cases} \frac{\|\boldsymbol{z}\|_2^2}{2} & \|\boldsymbol{z}\|_2 \leq \lambda, \\ \frac{\lambda^2}{2} & \|\boldsymbol{z}\|_2 > \lambda. \end{cases}$$

The corresponding group hard thresholding function is given by

$$\boldsymbol{\Theta}^{\mathrm{MS}}(\boldsymbol{o}; \lambda) = \begin{cases} \boldsymbol{0}, & \|\boldsymbol{o}\|_2 \leq \lambda, \\ \boldsymbol{o}, & \|\boldsymbol{o}\|_2 > \lambda. \end{cases}$$

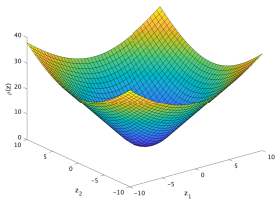
Multivariate Tukey

We define the multivariate version of Tukey's loss function as

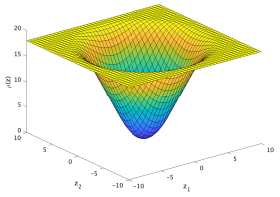
$$\rho^{\mathrm{MT}}_{\lambda}(\boldsymbol{z}) = \begin{cases} 1 - \left(1 - \frac{\|\boldsymbol{z}\|_2^2}{\lambda^2}\right)^3 & \|\boldsymbol{z}\|_2 \leq \lambda, \\ 1 & \|\boldsymbol{z}\|_2 > \lambda. \end{cases}$$

The corresponding group thresholding function can be expressed as

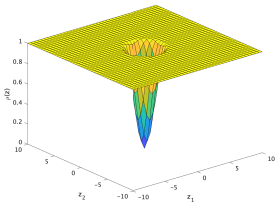
$$\Theta^{\mathrm{MT}}(\boldsymbol{o}; \lambda, \gamma) = \begin{cases} \boldsymbol{o} - \boldsymbol{o}\left(1 - \frac{\|\boldsymbol{o}\|_2^2}{\lambda^2}\right)^2 & \|\boldsymbol{o}\|_2 \leq \lambda, \\ \boldsymbol{o} & \|\boldsymbol{o}\|_2 > \lambda. \end{cases}$$



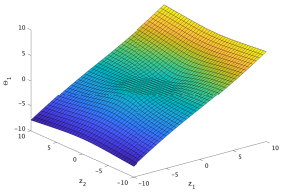
(a) multivariate Huber's loss



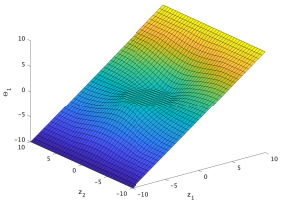
(b) group SCAD loss



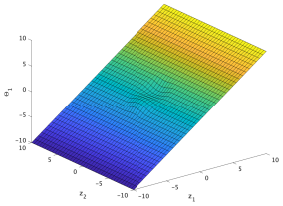
(c) group Tukey's loss



(d) group soft thresholding



(e) group SCAD thresholding



(f) group Tukey's thresholding

Figure 5.1: The multivariate loss functions (top row), the group-thresholding functions (bottom row). The x-axis and y-axis represent the values of input $\boldsymbol{z} \in \mathbb{R}^2$. The z-axis shows the output $\rho_{\lambda,\gamma}(\boldsymbol{z})$ in the top row, and the first component of $\Theta(\boldsymbol{z}; \lambda, \gamma)$ in the bottom row. The values of λ and γ are fixed with three.

Proposition 1 is inspired by similar propositions in She and Owen (2011) and Katayama and Fujisawa (2017). While they considered linear regression case and only the situation where $\boldsymbol{\Theta}(\cdot; \lambda, \gamma)$ is defined as the component-wise thresholding function, we consider clustering problem and the situation where $\Theta(\cdot; \lambda, \gamma)$ is a group-thresholding function. This enables us to solve clustering problems with multivariate robust loss functions by optimizing the problems with group penalties for outlier parameters instead.

5.3 Proposed method

5.3.1 Multi-task learning via robust regularized clustering

Almost all existing MTL methods based on clustering have not considered the presence of outlier tasks. For instance, MTLCVX shrinks the difference of centroids including those of outlier tasks, which contaminates the estimation of centroids. As a result, the estimation of regression coefficients concerning tasks corresponding to contaminated centroids also worsens. This motivates us to separate the estimation of the parameters concerning task clusters from outlier tasks. To the best of our knowledge, only Yao et al. (2019) consider the presence of outlier tasks and clustering of tasks simultaneously. However, their way of making the robustness of their method is not clear. On the other hand, some robust MTL methods (Chen et al., 2011; Gong et al., 2012) have attempted to address the issues of outlier tasks by introducing outlier parameters and selecting them using group lasso regularization. However, group lasso (Yuan and Lin, 2006) limits the value of the outlier parameters, which may not adequately represent their nature. To overcome these problems, we propose the **M**ulti-**T**ask **L**earning via **R**obust **R**egularized **C**lustering (MTLRRC). MTLRRC is formulated as follows:

$$\min_{\boldsymbol{w}_0,W,U,O} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \|\boldsymbol{w}_m - \boldsymbol{u}_m - \boldsymbol{o}_m\|_2^2 \right. \\ \left. + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 + \sum_{m=1}^{T} P(\boldsymbol{o}_m; \lambda_3, \gamma) \right\}, \tag{5.9}$$

where $\boldsymbol{o}_m = (o_{m1}, \ldots, o_{mp})^\top \in \mathbb{R}^p$ is a vector of outlier parameters for m-th task, λ_3 is a regularization parameter with a non-negative value. The second through the fourth term is based on the minimization problem (5.3). Then, if $\boldsymbol{o}_m$ is estimated to be a non-zero vector, m-th task is considered to be an outlier task that does not share a common structure with any tasks. We set the weights r_{m_1,m_2} based on Eq. (4.2)

In this problem, the centroid parameters $\boldsymbol{u}_m$ whose difference is shrunk and outlier

parameters $\boldsymbol{u}_m$ estimated to be zero vector or not are separated in the light of regularization. Thus, the cluster center component and the potential outlier component included in a task are estimated separately. Furthermore, if we employ non-convex regularization terms such as group SCAD, we can allow $\boldsymbol{o}_m$ to have a large value, which also leads to the large difference between $\boldsymbol{w}_m$ and $\boldsymbol{u}_m$. Then, the contamination from outlier tasks with significant unique characteristics is expected to be reduced. Next, we introduce another interpretation of the proposed method.

5.3.2 Interpretation through the BCD algorithm

5.3.2.1 Convex case

First, we consider that group lasso is employed for $P(\cdot; \lambda, \gamma)$ in (5.9). Then, since MTLRRC is a convex optimization problem, we can obtain another representation for (5.9) by minimizing in terms of O as follows:

$$\min_{\boldsymbol{w}_0, W, U} \left\{ \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 \sum_{m=1}^{T} h^{\mathrm{M}}_{\lambda_3/\lambda_1}(\boldsymbol{w}_m - \boldsymbol{u}_m) \right. \\ \left. + \lambda_2 \sum_{(m_1, m_2) \in \mathcal{E}} r_{m_1, m_2} \|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \right\}, \tag{5.10}$$

where $h^{\mathrm{M}}_{\lambda}(\cdot)$ is a multivariate Huber's loss function (Hampel et al., 1986) defined as

$$h^{\mathrm{M}}_{\lambda}(\boldsymbol{z}) = \begin{cases} \frac{1}{2}\|\boldsymbol{z}\|_2^2 & \|\boldsymbol{z}\|_2 \leq \lambda, \\ \lambda \|\boldsymbol{z}\|_2 - \frac{\lambda^2}{2} & \|\boldsymbol{z}\|_2 > \lambda. \end{cases}$$

This representation helps us understand the interpretation of the proposed method (5.9).

Algorithm 5 Block coordinate descent algorithm for Problem (5.10)

Require: $(\boldsymbol{y}_m, X_m; m = 1, \ldots, T), R, \lambda_1, \lambda_2, \lambda_3, U^{(0)}$

while until convergence of $W^{(t)}$ and $U^{(t)}$ **do**

$$(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}) = \arg\min_{\boldsymbol{w}_0, W} L^{\mathrm{MH}}(\boldsymbol{w}_0, W, U^{(t)}) \tag{5.11}$$

$$U^{(t+1)} = \arg\min_{U} L^{\mathrm{MH}}(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U) \tag{5.12}$$

end while

Ensure: W, U

Let the objective function of the minimization problem (5.10) be $L^{\mathrm{MH}}(\boldsymbol{w}_0, W, U)$. We consider solving the minimization problem (5.10) by Algorithm 5 based on the BCD algorithm. Since the minimization in terms of $\boldsymbol{w}_m$ is separable, the update (5.11) is expressed as

$$(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)}) = \arg\min_{w_0, \boldsymbol{w}_m} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \lambda_1 h^{\mathrm{M}}_{\lambda_3/\lambda_1}(\boldsymbol{w}_m - \boldsymbol{u}_m^{(t)}) \right\}, \quad m = 1, \ldots, T.$$

These updates estimate the regression coefficients for the m-th task to be close to the corresponding centroid. However, when $\|\boldsymbol{w} - \boldsymbol{u}_m^{(t)}\|_2$ tends to take a larger value than λ_3/λ_1, the shrinkage toward the centroid is reduced by the part of L_2-norm in the multivariate Huber's function. Thus, if m-th task is an outlier task, the estimated $\widehat{\boldsymbol{w}}_m$ is expected to be less affected by the common structure $\widehat{\boldsymbol{u}}_m$.

The minimization problem in terms of the update (5.12) is in the framework of the minimization problem (5.8). This can be seen by replacing $(\boldsymbol{x}_i; i = 1, \ldots, n)$ in (5.8) with $(\boldsymbol{w}_m^{(t+1)}; m = 1, \ldots, T)$ and $\rho_{\lambda_2,\gamma}(\cdot)$ with $h^{\mathrm{HM}}_{\lambda_3}(\cdot)$. Consequently, the update of $U^{(t+1)}$ is performed under the robust clustering of tasks. Based on the discussions of Algorithm 5, the estimated values $\widehat{W}$ and $\widehat{U}$ in MTLRRC can be regarded as a convergence point of alternative estimation, which consists of a regression step that reduces shrinkage of outlier tasks toward cluster center and a robust clustering step for tasks. Therefore,

MTLRRC is expected to be robust to the outlier tasks.

5.3.2.2 Non-convex case

Next, we consider that non-convex group penalties are employed for $P(\cdot;\lambda,\gamma)$ in (5.9). The estimates of the parameters can be calculated by Algorithm 6. Here, $L^{\mathrm{MR}}(\boldsymbol{w}_0, W, U, O)$ is the objective function of the minimization problem (5.9). Then, the following proposition similar to Proposition 1 holds.

Proposition 2. *Let* $\boldsymbol{w}'_m = (w_{m0}, \boldsymbol{w}_m^\top)^\top$. *Suppose that* $\left(\widehat{\boldsymbol{w}}_0, \widehat{W}, \widehat{U}\right)$ *is a pair of convergence point in Algorithm 6 and* $\boldsymbol{\psi}(\boldsymbol{o};\lambda,\gamma) = \boldsymbol{o} - \boldsymbol{\Theta}(\boldsymbol{o};\lambda,\gamma)$. *Then,* $(\widehat{\boldsymbol{w}}_0, \widehat{W}, \widehat{U})$ *satisfies*

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{w}'_m}\frac{1}{n_m}L(w_{m0},\boldsymbol{w}_m)|_{\boldsymbol{w}'_m=\widehat{\boldsymbol{w}}'_m} + \lambda_1 \begin{pmatrix} 0 \\ \boldsymbol{\psi}(\widehat{\boldsymbol{w}}_m - \widehat{\boldsymbol{u}}_m;\lambda_3/\lambda_1,\gamma) \end{pmatrix} = \boldsymbol{0}, \quad m = 1,\ldots,T, \\
&-\lambda_1\Psi(\widehat{W}-\widehat{U};\lambda_3/\lambda_1,\gamma) + \lambda_2\partial_{\mathrm{vec}(U)}(\|D_r\mathrm{vec}(U)\|_{2,1})|_{U=\widehat{U}} \ni \boldsymbol{0}.
\end{aligned} \tag{5.13}$$

The proof of the proposition is given by Section 5.8.2. When the $P(\boldsymbol{z};\lambda,\gamma)$ is employed such that corresponding $\rho_{\lambda,\gamma}(\boldsymbol{z})$ is a weakly convex function, the equations (5.13) are the same first-order conditions for the following minimization problem:

$$\begin{aligned}
\min_{\substack{\boldsymbol{w}_m,\boldsymbol{u}_m\in\mathbb{R}^p,\\ m=1,\ldots,T}} \Bigg\{ &\sum_{m=1}^T \frac{1}{n_m}L(w_{m0},\boldsymbol{w}_m) + \lambda_1\sum_{m=1}^T \rho_{\lambda_3/\lambda_1,\gamma}(\boldsymbol{w}_m - \boldsymbol{u}_m) \\
&+\lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2}\|\boldsymbol{u}_{m_1} - \boldsymbol{u}_{m_2}\|_2 \Bigg\}.
\end{aligned} \tag{5.14}$$

Therefore, we may expect that the solution in the case of non-convex penalties has a similar interpretation of the minimization problem (5.10).

5.4 Estimation algorithm via modified ADMM

MTLRRC can be estimated by Algorithm 6. However, this estimation procedure is computationally expensive, because the update of $U^{(t+1)}$ involves solving the convex

Algorithm 6 Block coordinate descent algorithm for MTLRRC

Require: $(\boldsymbol{y}_m, X_m; m = 1, \ldots, T), R, \lambda_1, \lambda_2, \lambda_3, U^{(0)}, O^{(0)}$

while until convergence of $\boldsymbol{w}_0^{(t)}, W^{(t)}$, $U^{(t)}$ and $O^{(t)}$ **do**

$$
\begin{aligned}
(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}) &= \arg\min_{\boldsymbol{w}_0, W} L^{\mathrm{MR}}(\boldsymbol{w}_0, W, U^{(t)}, O^{(t)}) \\
U^{(t+1)} &= \arg\min_{U} L^{\mathrm{MR}}(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U, O^{(t)}) \\
O^{(t+1)} &= \arg\min_{O} L^{\mathrm{MR}}(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O)
\end{aligned}
$$

end while

Ensure: $\boldsymbol{w}_0, W, U, O$

clustering, which is computationally demanding. To avoid this computation, we consider estimating parameters included in MTLRRC by alternating direction method of multipliers (ADMM; Boyd et al. (2011)).

We consider the following minimization problem equivalent to Problem (5.9):

$$
\begin{aligned}
\min_{\boldsymbol{w}_0, W, U, O} \Bigg\{ & \sum_{m=1}^{T} \frac{1}{n_m} L(\boldsymbol{w}_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \|W - U - O\|_F^2 \\
& + \lambda_2 \sum_{(m_1, m_2) \in \mathcal{E}} r_{m_1, m_2} \|\boldsymbol{b}_{(m_1, m_2)}\|_2 + \sum_{m=1}^{T} P(\boldsymbol{o}_m; \lambda_3, \gamma) \Bigg\}, \\
\text{s.t.} \quad & A_{\mathcal{E}} U = B.
\end{aligned}
$$

For this minimization problem, we consider the following augmented Lagrangian:

$$
\begin{aligned}
L_\nu(\boldsymbol{w}_0, W, U, O, B, S) = & \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \|W - U - O\|_F^2 \\
& + \lambda_2 \sum_{(m_1, m_2) \in \mathcal{E}} r_{m_1, m_2} \|\boldsymbol{b}_{(m_1, m_2)}\|_2 + \sum_{m=1}^{T} P(\boldsymbol{o}_m; \lambda_3, \gamma) \\
& + \mathrm{tr}(S^\top (B - A_{\mathcal{E}} U)) + \frac{\nu}{2} \|B - A_{\mathcal{E}} U\|_F^2,
\end{aligned} \tag{5.15}
$$

where $A_{\mathcal{E}}$ is a $|\mathcal{E}| \times T$ matrix whose each row is $\boldsymbol{a}_{m_1, m_2}^\top$ defined with the same manner

of Eq. (3.6), S is a $|\mathcal{E}| \times p$ Lagrangian multipliers matrix, and ν is a tuning parameter with non-negative value.

For this augmented Lagrangian, we consider the following updates of the modified ADMM in Chapter 3:

$$
\begin{aligned}
(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}) &= \arg\min_{\boldsymbol{w}_0, W} L_\nu(\boldsymbol{w}_0, W, U^{(t)}, O^{(t)}, B, S^{(t)}), \\
U^{(t+1)} &= \arg\min_{U} \left(\min_{B} L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U, O^{(t)}, B, S^{(t)}) \right), \\
O^{(t+1)} &= \arg\min_{O} L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B, S^{(t)}), \\
S_{(m_1,m_2)}^{(t+1)} &= \mathrm{prox}((S^{(t)} + \nu A_{\mathcal{E}} U^{(t+1)})_{(m_1,m_2)}, \lambda_2 r_{m_1,m_2}), \quad (m_1, m_2) \in \mathcal{E},
\end{aligned}
$$

The update for $\boldsymbol{w}_0$ and W is given by solving the independent regularized GLMs for each task, which are expressed as

$$
(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)}) = \arg\min_{w_{m0}, \boldsymbol{w}_m} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \|\boldsymbol{w}_m - \boldsymbol{u}_m^{(t)} - \boldsymbol{o}_m^{(t)}\|_2^2 \right\}, \ m = 1, \ldots, T.
$$

These minimization problems are solved by the Newton-Raphson method provided in Algorithm 3.

The minimization problem in terms of U and B is jointly done. The minimization in terms of B can be written explicitly as shown in Chapter 3. We only need to solve that in terms of U using the accelerated gradient method, which is provided in Algorithm 8.

The update of O is given by the same manner as (5.6). As a result, we obtain the estimation algorithm for MTLRRC as Algorithm 7.

Although the convergence of ADMM for non-convex functions has been shown in some studies (e.g. Wang et al. (2019) and Fan and Yin (2024)), the convergence of the proposed method is non-trivial because of the modification of the ADMM algorithm based on Shimmura and Suzuki (2022). We provide a theoretical guarantee regarding convergence to a limit point, which is summarized as the following theorem.

Theorem 1. *Assume that $P(\cdot; \lambda, \gamma)$ is a weakly convex function with modules C_p. If $\lambda_1 > C_p$ and $\nu > \frac{2\lambda_1^2}{\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})(\lambda_1 - C_p)}$ are satisfied, the sequence $\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\}$*

generated by the Algorithm 7 converges to a limit point $\{\widehat{\boldsymbol{w}}_0^{(*)}, \widehat{W}^{(*)}, \widehat{U}^{(*)}, \widehat{O}^{(*)}, \widehat{B}^{(*)}, \widehat{S}^{(*)}\}$. *In addition, any limit point is a stationary point of the augmented Lagrangian* (5.15).

The proof of the theorem is given by Section 5.8.2. This theorem ensures that our modified ADMM algorithm converges to a stationary point under mild conditions. The conditions provide practical guidelines for parameter selection in the algorithm. Specifically, we can ensure convergence by setting λ_1 larger than the weak convexity modulus C_p and choosing an appropriate step size ν. However, as we will see in later simulation studies, even if these conditions are not satisfied, the algorithm will converge empirically in many situations.

Furthermore, we consider the following augmented Lagrangian derived from Problem (5.14):

$$\begin{aligned} L_\nu(\boldsymbol{w}_0, W, U, O, B, S) = &\sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \sum_{m=1}^{T} \rho_{\lambda_3/\lambda_1,\gamma}(\boldsymbol{w}_m - \boldsymbol{u}_m) \\ &+ \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{b}_{(m_1,m_2)}\|_2 \\ &+ \mathrm{tr}(S^\top (B - A_\mathcal{E} U)) + \frac{\nu}{2} \|B - A_\mathcal{E} U\|_F^2. \end{aligned} \tag{5.16}$$

Then, the following holds.

Proposition 3. *Under the assumption of Theorem 1 with* $\lambda_1 > C_p$ *and* $\nu > \frac{2\lambda_1^2}{\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})(\lambda_1 - C_p)}$. *Then, any limit point* $\{\widehat{\boldsymbol{w}}_0^{(*)}, \widehat{W}^{(*)}, \widehat{U}^{(*)}, \widehat{B}^{(*)}, \widehat{S}^{(*)}\}$ *is one of the stationary points for the minimization problem concerning* (5.16).

The proof of the proposition is given by Section 5.8.2. This proposition is the ADMM version of Proposition 2, which would justify minimizing MTLRRC by using the modified ADMM algorithm instead of the BCD algorithm. Note that due to the nonconvexity of the objective function except when group lasso is used, there may be a duality gap between the original MTLRRC and the minimization problem concerning its augmented Lagrangian.

We analyze the computational complexity of our proposed method. The initialization of ι requires $\mathcal{O}(|\mathcal{E}|T^2)$ operations, which is computed only once before the ADMM iterations. Within each ADMM iteration, updating $(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)})$ via the Newton-Raphson method requires $\mathcal{O}(N_{\mathrm{NR}}p^2(n+Tp))$ operations, where N_{NR} denotes the number of Newton-Raphson iterations. The update of $U^{(t+1)}$ via accelerated gradient method requires $\mathcal{O}(N_{\mathrm{AG}}|\mathcal{E}|Tp)$ operations, where N_{AG} represents the number of accelerated gradient method iterations. The update of $O^{(t+1)}$ requires $\mathcal{O}(Tp)$ operations. The update of $S^{(t+1)}$ requires $\mathcal{O}(|\mathcal{E}|Tp)$ operations. Consequently, each iteration of Algorithm 7 has an overall computational complexity of $\mathcal{O}(N_{\mathrm{NR}}p^2(n+Tp) + N_{\mathrm{AG}}|\mathcal{E}|Tp)$. In contrast, Algorithm 6 requires $\mathcal{O}(N_{\mathrm{NR}}p^2(n+Tp) + N_{\mathrm{A}}N_{\mathrm{AG}}|\mathcal{E}|Tp)$ operations, where N_{A} denotes the number of ADMM iterations required for solving the convex clustering problem. Thus, the computational complexity of the BCD algorithm regarding the second term is N_{A} times larger than that of the modified ADMM algorithm, which can be substantial when N_{A} is large.

We further compare our method with RCMTL (Yao et al., 2019), which is the only existing method that addresses both robustness against outlier tasks and task clustering. RCMTL employs a BCD algorithm incorporating two inner ADMM algorithms. Their algorithm requires $\mathcal{O}(N_{\mathrm{A}_1}N_{\mathrm{G}}Tp(n+T) + N_{\mathrm{A}_2}T^2p)$ operations in each iteration, where N_{A_1}, N_{A_2} , and N_{G} denote the iteration counts for the first inner ADMM, second inner ADMM, and gradient method within the first inner ADMM, respectively. In comparing MTLRRC and RCMTL, the first term in both complexity expressions primarily reflects the computational cost of updating regression coefficients, while the second term corresponds to updating the task relationship components. For the first term, direct comparison is difficult as RCMTL employs backtracking to avoid matrix inversion computations. Regarding the second term in our modified ADMM, when the weights r_{m_1,m_2} are constructed according to Eq. (4.2), the computational complexity becomes $\mathcal{O}(N_{\mathrm{AG}}kT^2p)$ since $|\mathcal{E}| \leq kT$. Given that k is typically small (e.g., $k = 5$)

in the context of convex clustering and considering the accelerated gradient method's quadratic convergence rate, our modified ADMM algorithm achieves computational efficiency comparable to or better than RCMTL in the task relationship part.

Algorithm 7 Estimation algorithm of MTLRRC via modified ADMM

Require: $\{\boldsymbol{y}_m, X_m; m = 1, \ldots, T\}, k, \lambda_1, \lambda_2, \lambda_3, \gamma, U^{(0)}, O^{(0)}$

for $m = 1, \ldots, T$ **do**

$\widehat{\boldsymbol{w}}_m^{\mathrm{STL}} = \mathrm{STL}(\boldsymbol{y}_m, X_m)$

end for

calculating R by Eq. (4.2) from k and $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$

converting R into $A_{\mathcal{E}}$ by Eq. (3.6)

$L = A_{\mathcal{E}}^{\top} A_{\mathcal{E}}, \iota = \frac{1}{\lambda_1 + 2\max_{i=1,\ldots,T}((L)_{ii})}$

while until convergence of $W^{(t)}$ **do**

update of $\boldsymbol{w}_0$ and W

for $m = 1, \ldots, T$ **do**

$(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)\top})^{\top} = \mathrm{NR}(n_m, X_m, \boldsymbol{y}_m, (\boldsymbol{u}_m^{(t)} + \boldsymbol{o}_m^{(t)}), \lambda_1)$

end for

update of U

$U^{(t+1)} = \mathrm{AGU}(W^{(t+1)}, O^{(t)}, S^{(t)})$

update of O

for $m = 1, \ldots, T$ **do**

$\boldsymbol{o}_m^{(t+1)} = \boldsymbol{\Theta}(\boldsymbol{w}_m^{(t+1)} - \boldsymbol{u}_m^{(t+1)}; \lambda_3/\lambda_1, \gamma)$

end for

update of S

for $(m_1, m_2) \in \mathcal{E}$ **do**

$S_{(m_1,m_2)}^{(t+1)} = \mathrm{prox}((S^{(t)} + \nu A_{\mathcal{E}} \cdot U^{(t+1)})_{(m_1,m_2)}, \lambda_2 r_{m_1,m_2})$

end for

end while

Ensure: $\boldsymbol{w}_0, W, U, O$

Algorithm 8 Update of U via accelerated gradient method

function AGU(W, O, S)

$l = 0, \alpha^{(0)} = 1, H^{(0)} = W, E^{(0)} = W$

while until convergence of $H^{(l)}$ **do**

for $(m_1, m_2) \in \mathcal{E}$ **do**

$F_{(m_1,m_2)} = \text{prox}((S + \nu A_{\mathcal{E}} \cdot E^{(l)})_{m_1,m_2}, \lambda_2 r_{m_1,m_2})$

end for

$H^{(l+1)} = E^{(l)} - \iota\{\lambda_1(E^{(l)} + O - W) + A_{\mathcal{E}}^{\top} F\}$

$\alpha^{(l+1)} = \frac{1+\sqrt{1+4(\alpha^{(l)})^2}}{2}$

$E^{(l+1)} = E^{(l)} + \frac{\alpha^{(l)}-1}{\alpha^{(l+1)}}(H^{(l+1)} - H^{(l)})$

end while

Output: $U = H^{(l)}$

end function

5.5 Simulation studies

In this section, we report simulation studies in the linear regression setting. We generated data by the true model:

$$\boldsymbol{y}_m = X_m \boldsymbol{w}_m^* + \boldsymbol{\epsilon}_m, \quad m = 1, \ldots, T,$$

where $\boldsymbol{\epsilon}_m$ is an error term whose each component is distributed as $N(0, \sigma^2)$ independently, X_m is a design matrix generated from $N_p(\mathbf{0}, I_p)$ independently, and $\boldsymbol{w}_m^*$ is a true regression coefficient vector for m-th task. For this true model, T tasks consist of C true clusters and other outlier tasks. First, all tasks were assigned to C clusters with the same number of tasks in each cluster as T/C. Then, some of them were randomly assigned to outlier tasks.

For the true structure of regression coefficient vectors, we considered the following

two cases:

$$\text{Case 1:} \quad \boldsymbol{w}_m^* = \boldsymbol{u}_c^* + \boldsymbol{v}_m^{c*} + I(\tau_m = 1)\boldsymbol{o}_m^{c*},$$

$$\text{Case 2:} \quad \boldsymbol{w}_m^* = \begin{cases} \boldsymbol{u}_c^* + \boldsymbol{v}_m^{c*} & \text{if } \tau_m = 0, \\ \boldsymbol{o}_m^* & \text{if } \tau_m = 1, \end{cases}$$

where $\boldsymbol{u}_c^*$ is a true cluster center for c-th cluster, $\boldsymbol{v}_m^{c*}$ is a true task-specific parameter for m-th task belonging to c-th cluster, $\boldsymbol{o}_m$ is a true outlier parameter for m-th task, and τ_m is a random variable distributed as $P(\tau_m = 1) = \kappa$ and $P(\tau_m = 0) = 1 - \kappa$. $\tau_m = 1$ means that the m-th task is assigned to an outlier task. Case 1 considers a situation where outlier tasks share the same cluster center with other tasks, but the outlier parameter is added. On the other hand, Case 2 considers a situation where outlier tasks do not have any common structure with other tasks.

The parameters $\boldsymbol{u}_c^*, \boldsymbol{v}_m^{c*}$, and $\boldsymbol{o}_m^*$ were generated as follows. First, each explanatory variable $\{j = 1, \ldots, p\}$ was randomly assigned to the c-th clusters $\{c = 1, \ldots, C\}$ with the same probability. Then, we generated a true centroid parameter for c-th cluster $\boldsymbol{u}_c^* = (u_{c1}^*, \ldots, u_{cp}^*)^\top$ by

$$u_{cj}^* \begin{cases} \sim N(0, 10) & \text{if } j\text{-th variable is assigned to } c\text{-th cluster}, \\ = 0 & \text{otherwise}, \end{cases} \qquad j = 1, \ldots, p.$$

Next, we generated a true task-specific parameter for m-th task that belongs to c-th cluster $\boldsymbol{v}_m^{c*} = (v_{m1}^{c*}, \ldots, v_{mp}^{c*})^\top$ by

$$v_{mj}^{c*} \begin{cases} \sim N(0, 1) & \text{if } j\text{-th variable is assigned to } c\text{-th cluster}, \\ = 0 & \text{otherwise}, \end{cases} \qquad j = 1, \ldots, p.$$

For Case 1, we generated a true outlier parameter for m-th task belonging to c-th cluster but assigned to an outlier task $\boldsymbol{o}_m^{c*} = (o_{m1}^{c*}, \ldots, o_{mp}^{c*})^\top$ by

$$o_{mj}^{c*} \begin{cases} \sim f^{\mathrm{MTN}}(o) & \text{if } j\text{-th variable is assigned to } c\text{-th cluster}, \\ = 0 & \text{otherwise}, \end{cases} \qquad j = 1, \ldots, p,$$

where $f^{\mathrm{MTN}}(o)$ is a mixture of truncated normal distribution given by

$$f^{\mathrm{MTN}}(o) = 0.5f(o, -\infty, -3, -3, \sigma_o^2) + 0.5f(o, 3, \infty, 3, \sigma_o^2),$$

where $f(o, a, b, \mu, \sigma)$ is a truncated normal distribution on $a \leq o \leq b$ whose original normal distribution has mean μ and variance σ^2. The reason for generating $o_{mj}^{(c)*}$ from $f^{\mathrm{MTM}}(o)$ is to leave the absolute value of $\boldsymbol{o}_{mj}^{c*}$ away from zero so that the outlier task is located away from the cluster. For Case 2, we generated a true outlier parameter for m-th task $\boldsymbol{o}_m^* = (o_{m1}^*, \ldots, o_{mp}^*)^\top$ by

$$o_{mj}^* \sim U(-10, 10), \quad j = 1, \ldots, p,$$

where $U(a, b)$ is the continuous uniform distribution. Note that $\boldsymbol{o}_m^{c*}$ in Case 1 shares the same index of non-zero variables with other tasks belonging to the c-th cluster, while all $\boldsymbol{o}_m^*$ in Case 2 do not have any zero variables.

From these generating ways of $\boldsymbol{u}_c^*$, $\boldsymbol{v}_m^{c*}$, $\boldsymbol{o}_m^{c*}$, and $\boldsymbol{o}_m^*$, true regression coefficient vectors for non-outlier tasks that belong to different clusters have different non-zero variables. In other words, tasks belonging to different clusters are orthogonal to each other.

For our true model, we set as $n_m = 200$, $p = 100$, $T = 150$, and $\sigma^2 = 5$. 200 samples in each task were split into 50 samples for the train, 100 samples for the validation, and left samples for the test. We considered settings: $\kappa = \{0, 0.1, 0.2, 0.3, 0.4\}$.

We compared MTLRRC with several methods for the evaluation. For MTLRRC, we consider the three cases where group lasso (GL), group SCAD (GS), and group MCP (GM) are used for the fourth term in (5.9). Here, we set $\gamma = 3.7$ for group SCAD and $\gamma = 3$ for group MCP. As other competing methods, we employed MTLCVX, MTLK (multi-task learning via k-means; Argyriou et al. (2007)), RCMTL (Yao et al., 2019), and Hotelling-like outlier task detection with MTLK (HMTLK). Here, HMTLK was done as follows. First, $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ $(m = 1, \ldots, T)$ were estimated and these sample mean $\bar{\boldsymbol{w}}^{\mathrm{STL}}$ and covariance matrix $\bar{\Sigma}^{\mathrm{STL}}$ were also calculated. For these values, we calculated the statistic $h_m = (\widehat{\boldsymbol{w}}_m^{\mathrm{STL}} - \bar{\boldsymbol{w}}^{\mathrm{STL}})^\top (\bar{\Sigma}^{\mathrm{STL}})^{-1} (\widehat{\boldsymbol{w}}_m^{\mathrm{STL}} - \bar{\boldsymbol{w}}^{\mathrm{STL}})$. Then, we detected tasks

that satisfied $h_m \geq \chi_p^{(95)}$, where $\chi_p^{(95)}$ is a 95 percentile point of χ^2 distribution having p degrees of freedom. Finally, we estimated regression coefficients by MTLK except for detected outlier tasks. Note that the detection based on Hotelling's T^2 is not theoretically justified for this simulation setting. However, as we will see later, it is possible to detect some outlier tasks.

The weights r_{m_1,m_2} for both MTLCVX and MTLRRC were calculated by Eq. (4.2). k was set to five. The estimation of $\widehat{\boldsymbol{w}}_m^{\mathrm{STL}}$ in Eq. (4.2) and HMTLK were performed by the lasso in R package "glmnet". In addition, the initialization of RCMTL was done by singular value decomposition of the initial regression coefficient matrix, which is also calculated by "glmnet". The tuning parameter ν included in Algorithm 7 was set to one. The regularization parameters were determined by the validation data. For the evaluation, we calculated the normalized mean squared error (NMSE), root mean squared error (RMSE), true positive rate (TPR), and false positive rate (FPR):

$$\begin{aligned}
\mathrm{NMSE} &= \frac{1}{T}\sum_{m=1}^{T}\frac{\|\boldsymbol{y}_m^* - X_m\widehat{\boldsymbol{w}}_m\|_2^2}{n_m \mathrm{Var}(\boldsymbol{y}_{\boldsymbol{m}}^*)}, \\
\mathrm{RMSE} &= \frac{1}{T}\sqrt{\sum_{m=1}^{T}\|\boldsymbol{w}_m^* - \widehat{\boldsymbol{w}}_m\|_2^2}, \\
\mathrm{TPR} &= \frac{\#\left\{m; \boldsymbol{o}_m^* \neq \boldsymbol{0} \ \wedge \ \widehat{\boldsymbol{o}}_m \neq \boldsymbol{0}\right\}}{\#\left\{m; \boldsymbol{o}_m^* \neq \boldsymbol{0}\right\}}, \\
\mathrm{FPR} &= \frac{\#\left\{m; \boldsymbol{o}_m^* = \boldsymbol{0} \ \wedge \ \widehat{\boldsymbol{o}}_m \neq \boldsymbol{0}\right\}}{\#\left\{m; \boldsymbol{o}_m^* = \boldsymbol{0}\right\}}.
\end{aligned}$$

NMSE and RMSE evaluate the accuracy of the prediction and estimated regression coefficients, respectively. TPR and FPR evaluate the accuracy of outlier detection. They were computed 40 times, and the mean and standard deviation were obtained in each setting.

Tables 5.1, 5.2, and 5.3 show the results of simulation studies for $\kappa = 0$, Cases 1 and 2, respectively. For Case 1, MTLRRC and MTLCVX show almost identical accuracy in NMSE and RMSE with all κs. MTLRRC, MTLCVX, and MTLK, which do not remove the outlier task a priori, outperform HMTLK in terms of NMSE and RMSE.

This may suggest that multi-task learning improves the estimation accuracy even for outlier tasks that are located away from other tasks. As for TPR, HMTLK shows better results than MTLRRC. If the purpose is only to detect and eliminate outlier tasks, HMTLK is probably a better choice than robust MTL methods. For FPR, MTLRRC with non-convex penalties archives almost the best performance with any κ. For Case 2, MTLRRC with non-convex penalties shows better performance than MTLCVX in terms of NMSE and RMSE. Furthermore, MTLRRC with non-convex penalties is superior to HMTLK in terms of both TPR and FPR. As for RCMTL, our results do not show any superior performance in all settings. This is possibly due to the incompatibility between initialization values obtained from "glmnet" and their parameter initialization method based on singular value decomposition, which causes the optimization algorithm to fall into local optima with poor accuracy. Note that in Tables 5.2 and 5.3, we do not highlight any TPR of RCMTL regardless of the large value, because RCMTL does not select almost all non-outlier tasks, and the value is meaningless.

On the whole, MTLRRC with non-convex penalties detected true outlier tasks while greatly minimizing the detection of false outlier tasks for both Case 1 and Case 2. However, the differences in estimation accuracy between MTLCVX and MTLRRC are small, particularly for small κ. On the other hand, MTLRRC with the group lasso regularization shows poor performance even for TPR and FPR. For outlier task detection, the group lasso regularization would not be recommended.

Table 5.1: Simulation result with non-outlier tasks

κ	Method	NMSE	RMSE	TPR	FPR
0	MTLRRC (GL)	**0.010** (0.003)	0.424 (0.073)	N/A	0.206 (0.358)
	MTLRRC (GS)	**0.010** (0.004)	**0.417** (0.073)	N/A	**0.001** (0.002)
	MTLRRC (GM)	0.011 (0.003)	0.424 (0.076)	N/A	**0.001** (0.003)
	RCMTL	0.750 (0.015)	4.040 (0.228)	N/A	1.000 (0.002)
	HMTLK	0.080 (0.061)	1.302 (0.469)	N/A	0.090 (0.080)
	MTLCVX	**0.010** (0.003)	0.427 (0.076)	–	–
	MTLK	0.029 (0.044)	0.603 (0.377)	–	–

Table 5.2: Simulation result of Case 1

κ	Method	NMSE	RMSE	TPR	FPR
0.1	MTLRRC (GL)	**0.035** (0.008)	1.056 (0.111)	0.386 (0.443)	0.263 (0.414)
	MTLRRC (GS)	**0.035** (0.007)	1.043 (0.120)	0.345 (0.434)	**0.001** (0.002)
	MTLRRC (GM)	0.036 (0.008)	1.048 (0.120)	0.363 (0.416)	**0.001** (0.004)
	RCMTL	0.747 (0.016)	4.152 (0.263)	0.998 (0.010)	1.000 (0.002)
	HMTLK	0.112 (0.039)	1.864 (0.314)	**0.724** (0.152)	0.073 (0.061)
	MTLCVX	**0.035**(0.009)	**1.030** (0.131)	–	–
	MTLK	0.054 (0.040)	1.190 (0.274)	–	–
0.2	MTLRRC (GL)	0.063 (0.012)	1.459 (0.136)	0.249 (0.356)	0.094 (0.279)
	MTLRRC (GS)	0.069 (0.015)	1.490 (0.146)	0.261 (0.381)	**0.001** (0.003)
	MTLRRC (GM)	0.066 (0.012)	1.459 (0.154)	0.352 (0.427)	**0.001** (0.004)
	RCMTL	0.744 (0.015)	4.349 (0.250)	1.000 (0)	1.000 (0)
	HMTLK	0.145 (0.040)	2.234 (0.290)	**0.693** (0.080)	0.040 (0.049)
	MTLCVX	**0.062** (0.014)	**1.437** (0.145)	–	–
	MTLK	0.079 (0.044)	1.528 (0.258)	–	–
0,3	MTLRRC (GL)	0.092 (0.019)	1.778 (0.140)	0.381 (0.426)	0.200 (0.359)
	MTLRRC (GS)	0.092 (0.020)	**1.761** (0.154)	0.149 (0.255)	**0.001** (0.003)
	MTLRRC (GM)	0.095 (0.017)	1.823 (0.143)	0.453 (0.440)	**0.001** (0.004)
	RCMTL	0.748 (0.016)	4.487 (0.265)	1.000 (0)	1.000 (0.001)
	HMTLK	0.189 (0.045)	2.596 (0.233)	**0.622** (0.078)	0.023 (0.025)
	MTLCVX	**0.091** (0.013)	1.778 (0.113)	–	–
	MTLK	0.106 (0.049)	1.836 (0.298)	–	–
0,4	MTLRRC (GL)	0.120 (0.020)	2.043 (0.122)	0.306 (0.382)	0.179 (0.333)
	MTLRRC (GS)	**0.117** (0.020)	2.035 (0.154)	0.317 (0.394)	**0.001** (0.003)
	MTLRRC (GM)	**0.117** (0.017)	**2.026** (0.117)	0.313 (0.383)	0.002 (0.004)
	RCMTL	0.746 (0.015)	4.552 (0.245)	1.000 (0)	1.000 (0)
	HMTLK	0.244 (0.044)	2.929 (0.196)	**0.586** (0.092)	0.036 (0.049)
	MTLCVX	0.122(0.020)	2.089 (0.139)	–	–
	MTLK	0.135 (0.056)	2.090 (0.288)	–	–

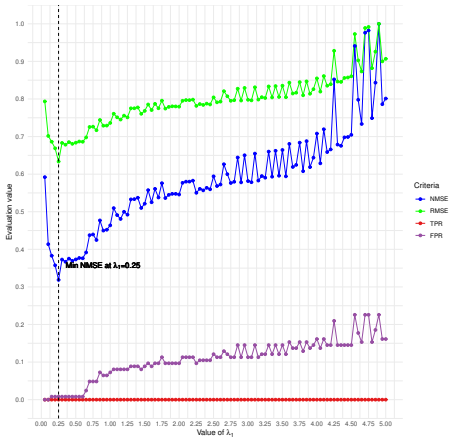
Table 5.3: Simulation result of Case 2

κ	Method	NMSE	RMSE	TPR	FPR
0.1	MTLRRC (GL)	0.069 (0.015)	1.205 (0.141)	0.294 (0.337)	0.304 (0.446)
	MTLRRC (GS)	0.069 (0.015)	1.202 (0.130)	**0.886** (0.308)	0.014 (0.013)
	MTLRRC (GM)	**0.067** (0.014)	**1.188** (0.132)	0.826 (0.358)	**0.008** (0.008)
	RCMTL	0.745 (0.012)	4.002 (0.257)	0.998 (0.016)	0.999 (0.003)
	HMTLK	0.157 (0.060)	1.867 (0.388)	0.613 (0.134)	0.111 (0.078)
	MTLCVX	0.072 (0.017)	1.233 (0.163)	–	–
	MTLK	0.094 (0.035)	1.380 (0.236)	–	–
0.2	MTLRRC (GL)	0.130 (0.019)	1.670 (0.135)	0.480 (0.361)	0.339 (0.414)
	MTLRRC (GS)	0.125 (0.021)	1.631 (0.136)	**0.936** (0.215)	**0.020** (0.019)
	MTLRRC (GM)	**0.118** (0.016)	**1.595** (0.115)	0.935 (0.224)	0.023 (0.013)
	RCMTL	0.740 (0.013)	4.053 (0.236)	1.000 (0)	0.999 (0.002)
	HMTLK	0.211 (0.044)	2.155 (0.251)	0.524 (0.100)	0.075 (0.066)
	MTLCVX	0.123 (0.020)	1.625 (0.139)	–	–
	MTLK	0.152 (0.035)	1.793 (0.201)	–	–
0.3	MTLRRC (GL)	0.173 (0.017)	1.935 (0.103)	0.467 (0.352)	0.317 (0.387)
	MTLRRC (GS)	0.171 (0.018)	1.914 (0.102)	**0.936** (0.226)	**0.020** (0.020)
	MTLRRC (GM)	**0.166** (0.018)	**1.881** (0.114)	0.911 (0.246)	0.021 (0.020)
	RCMTL	0.737 (0.014)	4.048 (0.184)	1.000 (0)	1.000 (0)
	HMTLK	0.276 (0.042)	2.459 (0.198)	0.542 (0.089)	0.066 (0.051)
	MTLCVX	0.181 (0.024)	1.972 (0.132)	–	–
	MTLK	0.204 (0.037)	2.081 (0.179)	–	–
0.4	MTLRRC (GL)	0.232 (0.022)	2.246 (0.109)	0.551 (0.387)	0.292 (0.391)
	MTLRRC (GS)	**0.220** (0.024)	**2.181** (0.115)	**0.943** (0.202)	**0.029** (0.025)
	MTLRRC (GM)	0.226 (0.028)	2.210 (0.134)	0.890 (0.275)	0.031 (0.023)
	RCMTL	0.736 (0.010)	4.003 (0.177)	1.000 (0.003)	1.000 (0.002)
	HMTLK	0.337 (0.049)	2.698 (0.199)	0.482 (0.078)	0.047 (0.043)
	MTLCVX	0.232 (0.024)	2.243 (0.122)	–	–
	MTLK	0.285 (0.047)	2.471 (0.196)	–	–

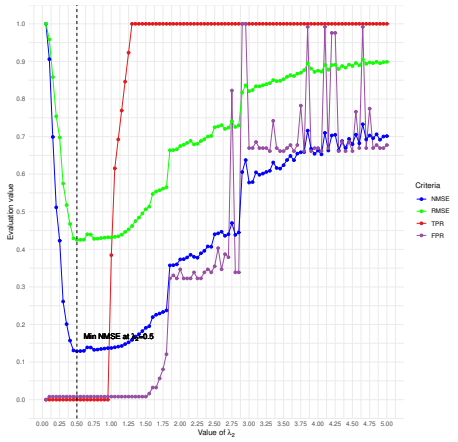
We conducted a sensitivity analysis for the selection of regularization parameters and γ. The analysis used a fixed dataset generated from Case 1 and Case 2 with $\kappa = 0.2$, maintaining other settings consistent with our previous simulation studies. For the regularization term, we employed the group SCAD penalty. First, we selected optimal regularization parameters except for fixed $\gamma = 3.7$ through grid search. Then, we estimated MTLRRC and calculated the evaluation values by varying only one parameter while fixing the other parameters to their optimal values. Consequently, we obtain curves representing the empirical relationship between each tuning parameter and the evaluation values as shown in Figures 5.2 and 5.3 for Case 1 and Case 2, respectively.

For λ_1, the selected value is the same for both Case 1 and Case 2. Only in Case 2, the selection of λ_1 is sensitive to all evaluation values. Since the trends of all evaluation values are synchronized, selecting the optimal value that minimizes NMSE yields favorable TPR and FPR values. Therefore it would be essential to search λ_1 in detail. For λ_2, in Case 1, while the curves of NMSE, RMSE and FPR are synchronized, that of TPR is not. This would explain the small TPR in the Case 1 results, as seen in Table 5.2. In Case 2, the evaluation values are consistent and less sensitive to the value of λ_2. For λ_3, in Case 1, there is not better λ_3 improving both FPR and TPR. The reason for this is that the appropriate λ_2 is not selected, and the search for the true structure regarding outlier tasks and cluster structure has failed. On the other hand, in Case 2, when λ_3 is larger than a certain value, TPR and NMSE are worsened simultaneously. This result probably coincides with our motivation to select outliers, which improves estimation accuracy. For γ, its value has a limited impact when other tuning parameters are appropriately selected. On the whole, if the outlier task has large characteristics, as in Case 2, then the proposed method can appropriately detect the outlier task by choosing parameters that minimize the NMSE. However, if the latent outlier components are subtle, as in Case 1, then selecting the parameters based on NMSE may fail to identify the outlier task. This limitation requires further investigation in

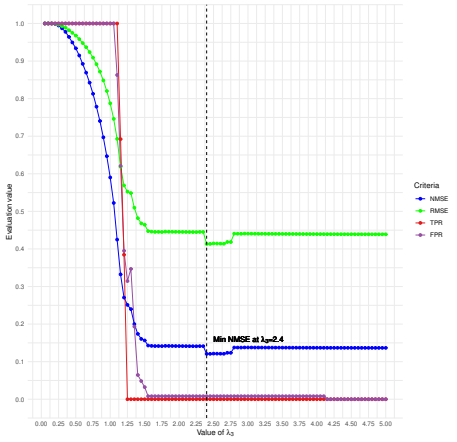
future work.



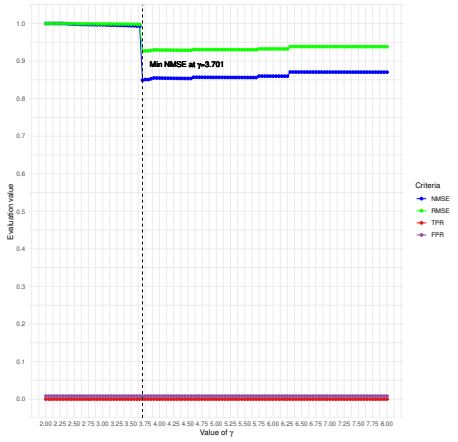
(a) λ_1 against evaluation values.



(b) λ_2 against evaluation values.

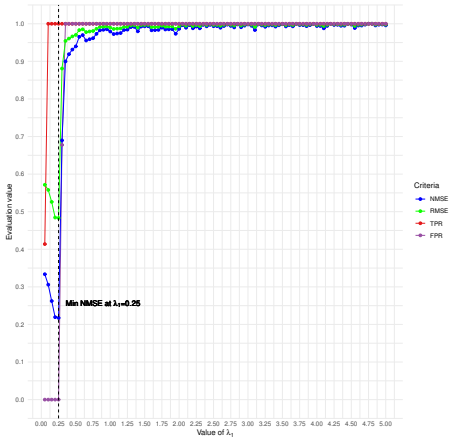


(c) λ_3 against evaluation values.

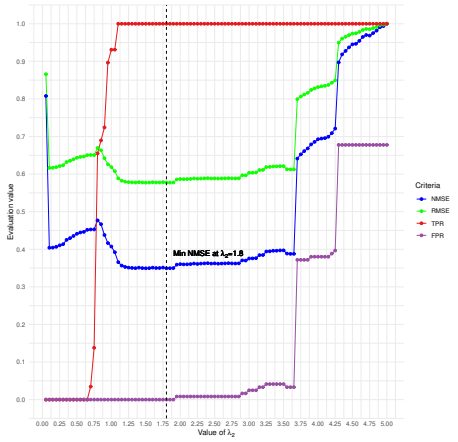


(d) γ against evaluation values.

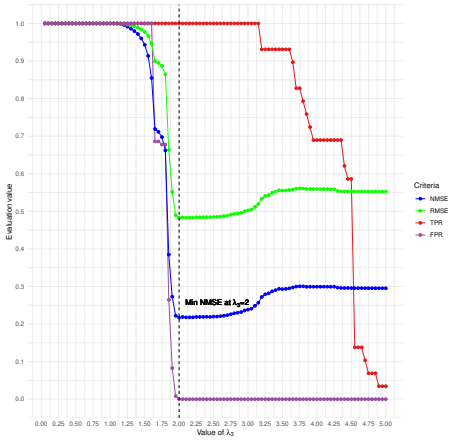
Figure 5.2: Each figure shows the relationship between one tuning parameter and evaluation values. The horizontal axis represents the parameter value, and the vertical axis represents the evaluation values. The blue line represents NMSE, the green line represents RMSE, the red line represents TPR, and the purple line represents FPR. Note that the RMSEs are normalized by those maximum values. The dashed line corresponds to the optimal value of the parameter that minimizes NMSE.



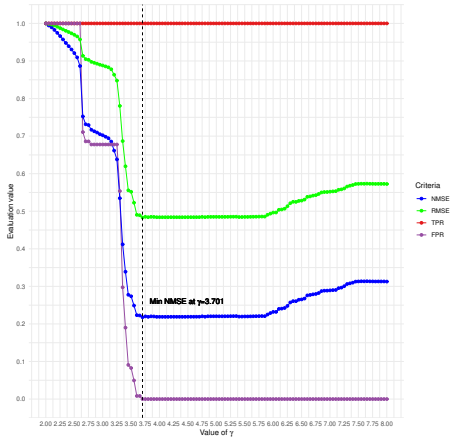
(a) λ_1 against evaluation values.



(b) λ_2 against evaluation values.



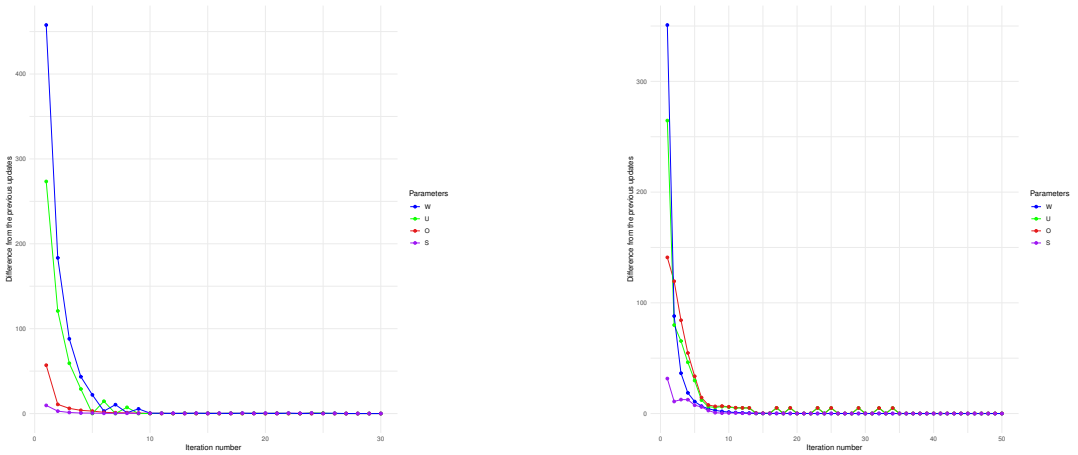
(c) λ_3 against evaluation values.



(d) γ against evaluation values.

Figure 5.3: Evaluation values against regularization parameters under Case 2.

We also check the convergence of our modified ADMM algorithm with group SCAD penalty. Figure 5.4 shows the results of the convergence under Case 2. Because the C_p of group SCAD is $\frac{1}{\gamma-1}$, $\lambda_1 > 0.371$ satisfies $\lambda_1 > C_p$ for $\gamma = 3.7$. The figure shows that all parameters converge under forty iterations for both $\lambda_1 > C_p$ and $\lambda_1 < C_p$.



(a) Convergence for $\lambda_1 = 0.05$.

(b) Convergence for $\lambda_1 = 0.5$

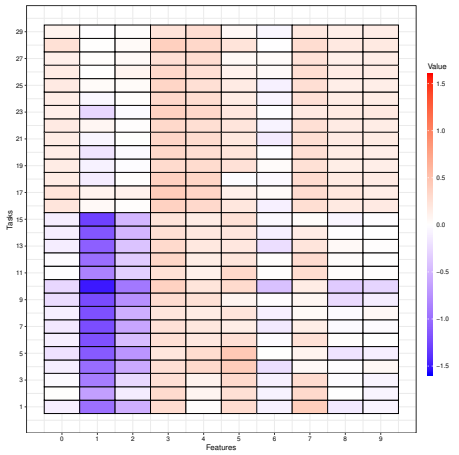
Figure 5.4: The horizontal axis represents the number of iterations (t). Vertical axis shows the difference of parameter values from previous iterations as blue line $\|W^{(t)} - W^{(t+1)}\|_F$, green line $\|U^{(t)} - U^{(t+1)}\|_F$, red line $\|O^{(t)} - O^{(t+1)}\|_F$, and purple line $\|S^{(t)} - S^{(t+1)}\|_F$, respectively.

5.6 Application to real datasets

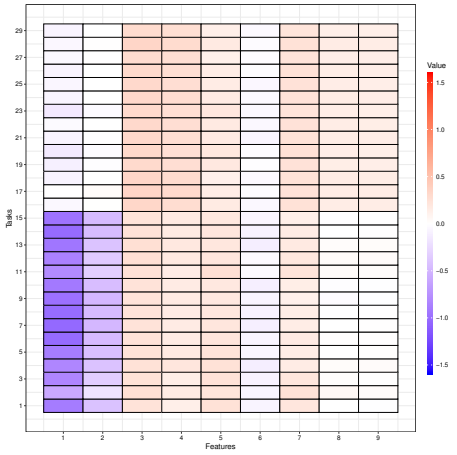
In this section, we apply MTLRRC to three real datasets. The first and second datasets are the landmine data (Xue et al., 2007) and school data (Bakker and Heskes, 2003), respectively. Similar to the previous chapter, the down-sampling is done for the landmine data. The third dataset is microarray data (Wille et al., 2004), which consists of microarray gene expression data focusing on isoprenoid biosynthesis in plants. The dataset contains expression levels of 21 genes in the mevalonate pathway and expression levels of 18 genes in the plastidial pathway. We used those 21 genes as feature variables and each of the 18 genes as a task. Since the dataset consists of only one design matrix, the setting is rather multivariate regression.

Table 5.4: AUC and NMSE for landmine data and school data in 100 repetitions

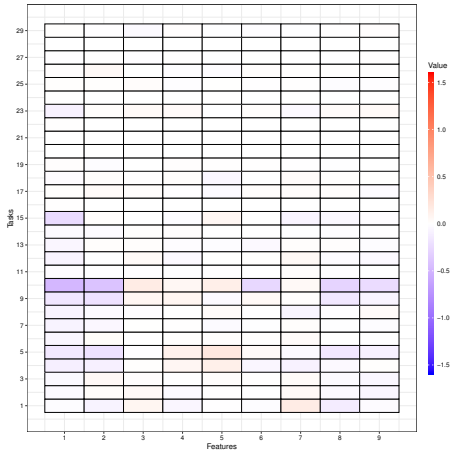
	Landmine	School	Microarray
Method	AUC	NMSE	NMSE
MTLRRC (GS)	**0.764** (0.021)	0.853 (0.058)	0.700 (0.064)
MTLRRC (GSγ)	**0.764** (0.024)	**0.844** (0.049)	**0.686** (0.066)
MTLRRC (GM)	0.760 (0.026)	0.852 (0.053)	0.691 (0.067)
MTLRRC (GMγ)	0.761 (0.026)	0.852 (0.058)	0.687 (0.066)
RCMTL	0.749 (0.022)	5.040 (0.256)	0.788 (0.069)
MTLCVX	0.760 (0.022)	0.847 (0.048)	0.694 (0.068)
DTFLR	0.704 (0.023)	-	0.724 (0.070)
MTLK	0.756 (0.023)	0.847 (0.048)	0.693 (0.065)



(a) Mean of estimated $\widehat{W}$ for 100 repetitions in the landmine data

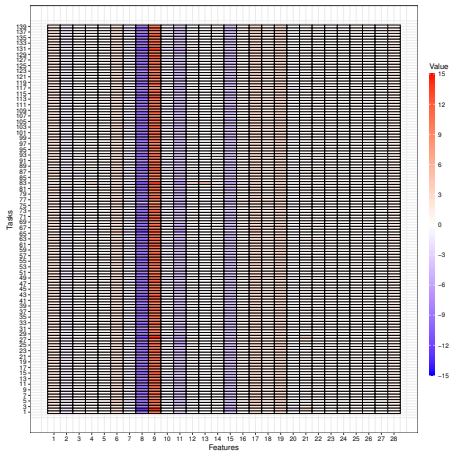


(b) Mean of estimated $\widehat{U}$ for 100 repetitions in the landmine data

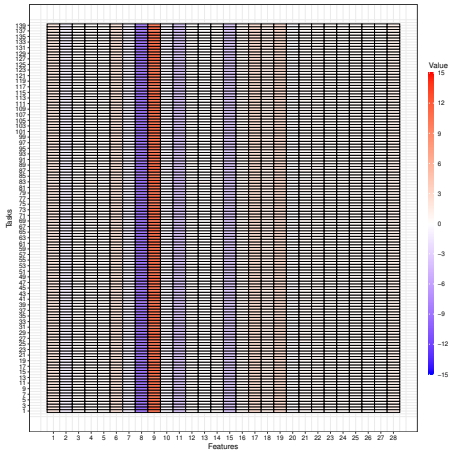


(c) Mean of estimated $\widehat{O}$ for 100 repetitions in the landmine data

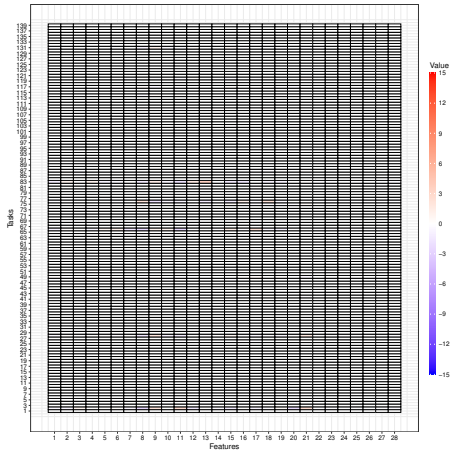
Figure 5.5: The mean of the estimated value of parameters in MTLRRC (GSγ) in 100 repetitions for the landmine data



(a) Mean of estimated $\widehat{W}$ for 100 repetitions in the school data

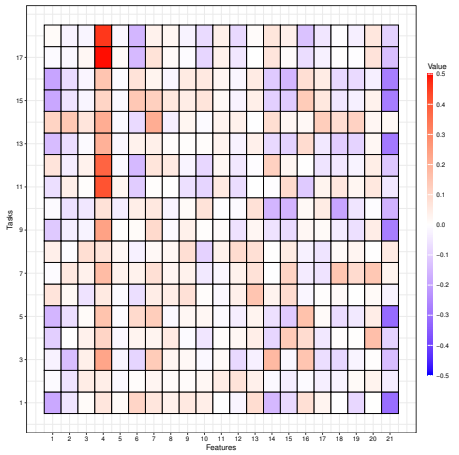


(b) Mean of estimated $\widehat{U}$ for 100 repetitions in the school data

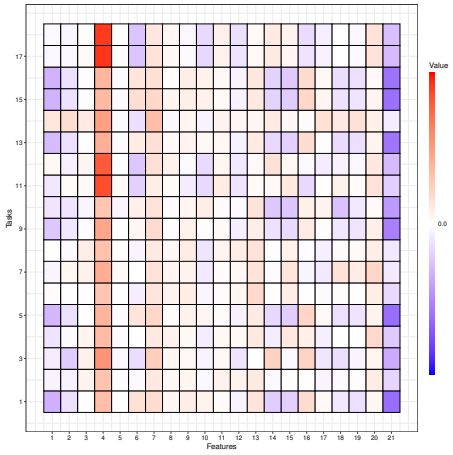


(c) Mean of estimated $\widehat{O}$ for 100 repetitions in the school data

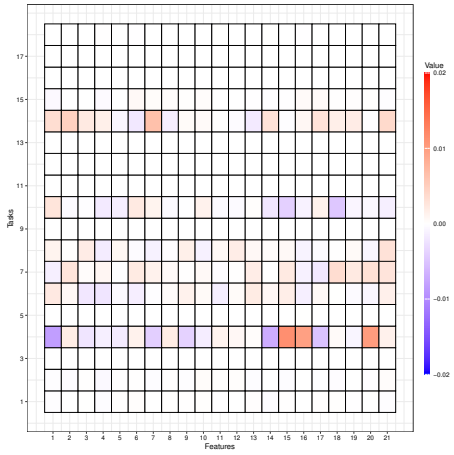
Figure 5.6: The mean of the estimated value of parameters in MTLRRC (GSγ) in 100 repetitions for the school data



(a) Mean of estimated $\widehat{W}$ for 100 repetitions in the microarray data



(b) Mean of estimated $\widehat{U}$ for 100 repetitions in the microarray data



(c) Mean of estimated $\widehat{O}$ for 100 repetitions in the microarray data

Figure 5.7: The mean of the estimated value of parameters in MTLRRC (GSγ) in 100 repetitions for the microarray data

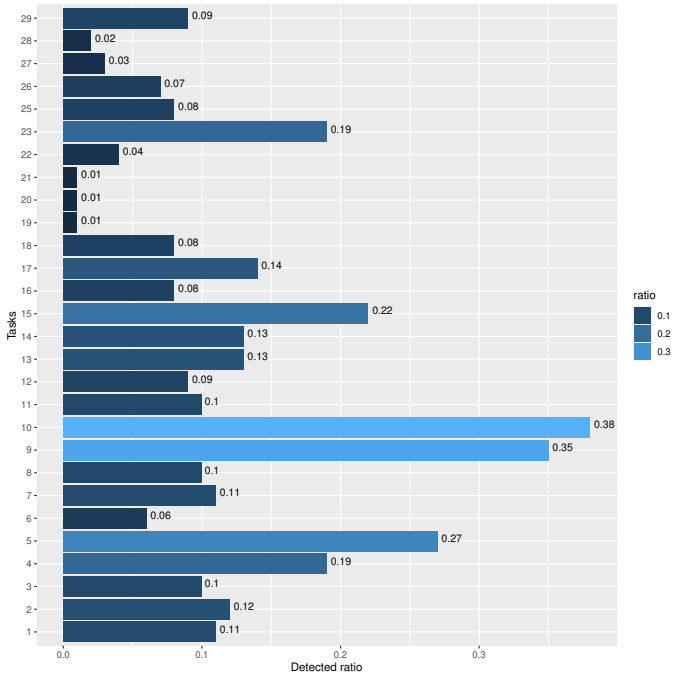


Figure 5.8: Ratio of $\widehat{\boldsymbol{o}}_m \neq \mathbf{0}$ for 100 repetitions in the landmine data

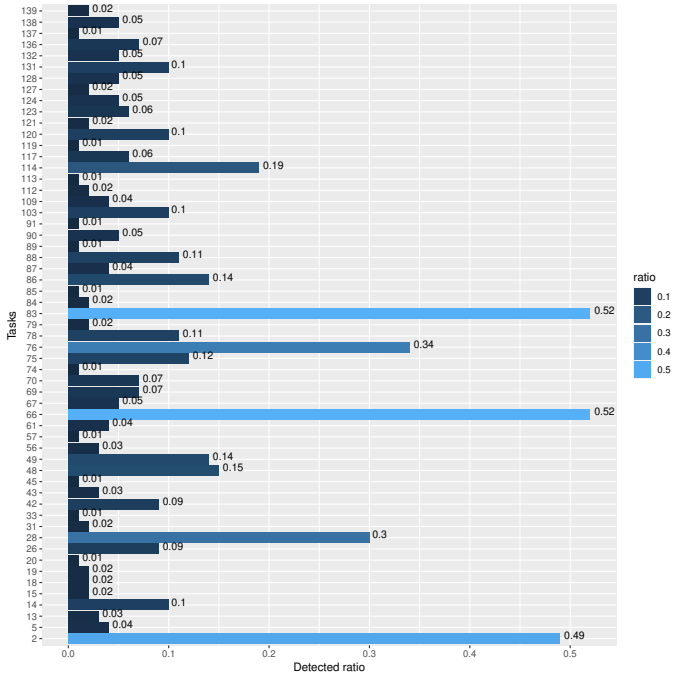


Figure 5.9: Ratio of $\widehat{\boldsymbol{o}}_m \neq \mathbf{0}$ for 100 repetitions in the school data

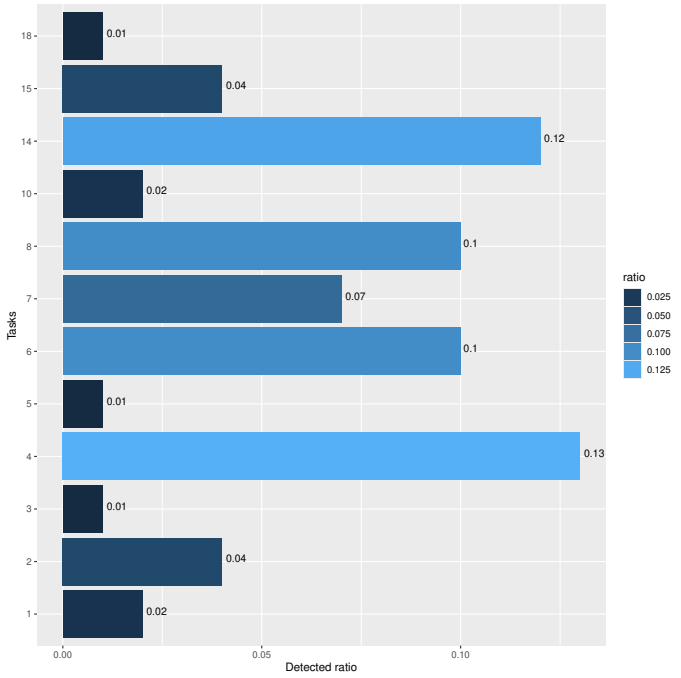


Figure 5.10: Ratio of $\widehat{\boldsymbol{o}}_m \neq \mathbf{0}$ for 100 repetitions in the microarray data

We compared MTLRRC with MTLCVX, RCMTL, DTFLR (distributed spanning-tree-based fused-lasso regression; Zhang et al. (2024)), and MTLK. First, we split the samples in each task into 60% for the train, 20% for the validation, and 20% for the test. The regularization parameters are determined by the validation data. In MTLRRC, we use non-convex penalties and consider four cases. In the first case, γ is fixed with 3 for the group MCP (GM). In the second case, γ is fixed with 3.7 for the group SCAD (GS). In the third and fourth cases, γ is chosen by the validation data for group MCP and group SCAD, respectively (GMγ and GSγ). For the evaluation, we calculated NMSE for the school data and the microarray data and AUC for the landmine data, respectively. These values were calculated 100 times with the random splitting of the dataset.

Table 5.4 shows the mean and standard deviation in 100 repetitions. Due to memory restrictions, the implementation code published by Zhang et al. (2024) was infeasible for the school data. From this table, we can observe that MTLRRC (GSγ) gives a smaller

NMSE and larger AUC than other methods for the school data, the microarray data, and the landmine data, respectively.

Next, we calculated the mean of estimated parameters $\widehat{W}, \widehat{U}$, and $\widehat{O}$ in MTL-RRC (GSγ). Figures 5.5 and 5.6 show the mean of estimated parameter values for the landmine data and the school data, respectively. The vertical axis is the index of tasks, and the horizontal axis is the index of features. Each color shows the mean of the estimated parameters. Figures 5.8 and 5.9 are bar plots that show the ratios of each task detected as an outlier task. Here, note that tasks that have never been detected are removed from the bar plots.

For the landmine data, Figure 5.5(b) suggests the presence of two clusters, which is consistent with the fact that tasks 1–15 and 16–29 are obtained from regions corresponding to different surface conditions. For outliers detection in Figure 5.5(c), because the 10-th task has a relatively larger or smaller value than other tasks and the task is a task detected as an outlier with a ratio greater than 0.3 from Figure 5.8, it may indicate that the task is a potential outlier task. Furthermore, the 9-th task also has a relatively high detection ratio that is greater than 0.3, although the mean of $\widehat{\boldsymbol{o}}_m$ is not as clear as the 10-th task from Figure 5.5(c). The estimated regression coefficients for the 9-th and 10-th tasks in Figure 5.5(a) show a similarity to other tasks within the same estimated cluster. These results may suggest the underlying structure among tasks in the landmine data is rather Case 1 than Case 2 in Section 5.5. Furthermore, from Figure 5.8, we observe that only two tasks in tasks 16–29 have an outlier task ratio of more than 0.1, while 13 tasks in tasks 1–15 have. This result suggests that the cluster composed of tasks 1–15 may have relatively large variability. This may provide some insight into the structure concerning sub-groups within the cluster.

For the school data, we obtained the homogeneous pattern in Figure 5.6(b). The school data have been considered rather homogeneous in some studies (Bakker and Heskes (2003); Evgeniou et al. (2005)). Thus, this result would be reasonable and

shows that MTLRRC can consistently estimate cluster structure even when the true number of underlying clusters is one. From Figure 5.9, the 2nd, 28th, 66th, 76th, and 83rd tasks were detected with a rate greater than 0.3. In addition, these tasks show larger values in $\boldsymbol{o}_m$ than other tasks from Figure 5.6(c). For the regression coefficients in Figure 5.6(a), although these tasks share many characteristics with other tasks, some regression coefficients have different characteristics. For example, the regression coefficients for 76th and 83rd tasks of 16th feature have, respectively, relatively large and small values that are much different from other tasks. For the microarray data, the analysis revealed heterogeneous patterns, as shown in Figures 5.7(a, b), without exhibiting distinctive features. Figure 5.10 indicates that none of the tasks were clearly identified as outliers, suggesting the absence of outlier tasks in the microarray dataset. This finding presents an interesting contrast to the school data and landmine data, where the existence of outlier tasks was indicated.

5.7 Discussion

We conducted simulation studies to evaluate the performance of MTLRRC under two scenarios of outlier task structures. In Case 1, outlier tasks share the same characteristics as the centroid but have additional outlier parameters. In Case 2, outlier tasks do not share any common structure and are independent of other tasks. In both cases, the proposed method with non-convex group penalties exhibited a near-zero FNR in outlier detection and a much larger TPR in Case 2. However, the improvement in estimation and prediction accuracy was slight when the proportion of outlier tasks was small.

In the application to real data, we observed that the proposed method effectively estimates multiple cluster structures and identifies potential outlier tasks resembling Case 1. These findings suggest that MTLRRC not only estimates clusters but also provides insights into the heterogeneity of outlier tasks within clusters. In the usual convex

clustering for observed and fixed data, one may look at the behavior of the cluster from the solution path. On the other hand, our clustering targets are regression coefficients, whose values also depend on the other regularization parameters like λ_1. Therefore, it is difficult to see the solution path. However, the outlier task detection enables us to obtain information about the behavior of the clusters at the determined regularization parameters. Furthermore, the circumstances corresponding to detected tasks can be subject to re-examination, because the dataset would have unique characteristics compared with other datasets.

One limitation of our study is that MTLRRC includes three or four regularization parameters to be determined. The computational cost of searching for the optimal value of these parameters can be demanding. On the other hand, the definition of the multivariate M-estimator having a connection to the group penalties is different from that of the traditional one (Maronna, 1976): while the former M-estimator is defined as a straightforward extension of the univariate M-estimator with robust multivariate loss function, the traditional one is defined as the solution of weighted log-likelihood equations. Although there may be some relationship between these two definitions, they are probably not equivalent. Moreover, outlier tasks similar to Case 2 were not detected in the analyzed real data. We leave these topics as future work.

5.8 Proofs

In this section, we provide some proof in this chapter.

5.8.1 Proofs of the weakly convexity

First, we set $h(\boldsymbol{z}) = \rho^{\mathrm{gSCAD}}_{\lambda,\gamma}(\boldsymbol{z}) + \frac{1}{2(\gamma-2)}\|\boldsymbol{z}\|_2^2$. The Hessian matrix in each region of $h(\boldsymbol{z})$ is semi-positive definite, as it can be easily verified. Next, we consider the limits of the derived function $\frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z})$ into an arbitrary point $\boldsymbol{z}_0$ on the boundary partitioning the

function from the inside and the outside of regions. The limits are given by

$$\lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \|\boldsymbol{z}\|_2\le\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \lambda\le\|\boldsymbol{z}\|_2<2\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \frac{\gamma-1}{\gamma-2}\boldsymbol{z}_0, \quad \text{s.t. } \|\boldsymbol{z}_0\|_2 = \lambda,$$

$$\lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \lambda\le\|\boldsymbol{z}\|_2<2\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ 2\lambda\le\|\boldsymbol{z}\|_2<\gamma\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \frac{\gamma}{2(\gamma-2)}\boldsymbol{z}_0, \quad \text{s.t. } \|\boldsymbol{z}_0\|_2 = 2\lambda,$$

$$\lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ 2\lambda\le\|\boldsymbol{z}\|_2<\gamma\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \gamma\lambda\le\|\boldsymbol{z}\|_2}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \frac{1}{\gamma-2}\boldsymbol{z}_0, \quad \text{s.t. } \|\boldsymbol{z}_0\|_2 = \gamma\lambda.$$

These imply that the derived function $\frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z})$ is continuous in the boundaries. Thus, $h(\boldsymbol{z})$ is a convex function and $\rho^{\mathrm{gSCAD}}_{\lambda,\gamma}(\boldsymbol{z})$ is a weakly convex function. Similarly, the group MCP is also a weakly convex function.

On the other hand, for the multivariate skipped mean loss, we set $h(\boldsymbol{z}) = \rho^{\mathrm{SM}}_{\lambda}(\boldsymbol{z}) + \frac{\delta}{2}\|\boldsymbol{z}\|_2^2$ ($\delta > 0$). The limits for the derived function into the boundary are calculated as follows:

$$\lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \|\boldsymbol{z}\|_2\le\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = (\delta+1)\boldsymbol{z}_0, \quad \text{s.t. } \|\boldsymbol{z}_0\|_2 = \lambda,$$

$$\lim_{\substack{\boldsymbol{z}\to\boldsymbol{z}_0\\ \|\boldsymbol{z}\|_2>\lambda}} \frac{\partial}{\partial \boldsymbol{z}} h(\boldsymbol{z}) = \delta\boldsymbol{z}_0, \quad \text{s.t. } \|\boldsymbol{z}_0\|_2 = \lambda,$$

Therefore, the derived function is discontinuous at the boundary. Then, multivariate skipped mean loss is not a weakly convex function.

5.8.2 Proofs of the propositions and theorem

Proof of Proposition 1. From the update of Algorithm 4, a convergence point $(\widehat{U}, \widehat{O})$ satisfies

$$\mathrm{vec}(\widehat{O}) = \begin{pmatrix} \widehat{\boldsymbol{o}}_1 \\ \widehat{\boldsymbol{o}}_2 \\ \vdots \\ \widehat{\boldsymbol{o}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Theta}(\boldsymbol{x}_1 - \widehat{\boldsymbol{u}}_1; \lambda_2, \gamma) \\ \boldsymbol{\Theta}(\boldsymbol{x}_2 - \widehat{\boldsymbol{u}}_2; \lambda_2, \gamma) \\ \vdots \\ \boldsymbol{\Theta}(\boldsymbol{x}_n - \widehat{\boldsymbol{u}}_n; \lambda_2, \gamma) \end{pmatrix}.$$

On the other hand, because the update (5.4) is equivalently expressed as

$$\widehat{U} = \arg\min_{U} \left\{ \frac{1}{2}\|\mathrm{vec}(X) - \mathrm{vec}(U) - \mathrm{vec}(\widehat{O})\|_2^2 + \lambda_1 \|D_r \mathrm{vec}(U)\|_{2,1} \right\},$$

it follows:

$$-\{\mathrm{vec}(X) - \mathrm{vec}(\widehat{U}) - \mathrm{vec}(\widehat{O})\} + \lambda_1 \partial_{\mathrm{vec}(U)} \|D_r \mathrm{vec}(U)\|_{2,1}|_{U=\widehat{U}} \ni \mathbf{0}.$$

Thus, we conclude Proposition 1 from

$$\begin{aligned}
&\mathrm{vec}(X) - \mathrm{vec}(\widehat{U}) - \mathrm{vec}(\widehat{O}) \\
&= \begin{pmatrix} \boldsymbol{x}_1 - \widehat{\boldsymbol{u}}_1 - \boldsymbol{\Theta}(\boldsymbol{x}_1 - \widehat{\boldsymbol{u}}_1; \lambda_2, \gamma) \\ \boldsymbol{x}_2 - \widehat{\boldsymbol{u}}_2 - \boldsymbol{\Theta}(\boldsymbol{x}_2 - \widehat{\boldsymbol{u}}_2; \lambda_2, \gamma) \\ \vdots \\ \boldsymbol{x}_n - \widehat{\boldsymbol{u}}_n - \boldsymbol{\Theta}(\boldsymbol{x}_n - \widehat{\boldsymbol{u}}_n; \lambda_2, \gamma) \end{pmatrix} \\
&= \begin{pmatrix} \psi(\boldsymbol{x}_1 - \widehat{\boldsymbol{u}}_1; \lambda_2, \gamma) \\ \psi(\boldsymbol{x}_2 - \widehat{\boldsymbol{u}}_2; \lambda_2, \gamma) \\ \vdots \\ \psi(\boldsymbol{x}_n - \widehat{\boldsymbol{u}}_n; \lambda_2, \gamma) \end{pmatrix} \\
&= \Psi(X - \widehat{U}; \lambda_2, \gamma).
\end{aligned}$$

□

Proof of Proposition 2. From Algorithm 6, a convergence point $(\widehat{\boldsymbol{w}}_0, \widehat{W}, \widehat{U}, \widehat{O})$ satisfies

$$\mathrm{vec}(\widehat{O}) = \begin{pmatrix} \widehat{\boldsymbol{o}}_1 \\ \widehat{\boldsymbol{o}}_2 \\ \vdots \\ \widehat{\boldsymbol{o}}_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Theta}(\widehat{\boldsymbol{w}}_1 - \widehat{\boldsymbol{u}}_1; \lambda_3/\lambda_1, \gamma) \\ \boldsymbol{\Theta}(\widehat{\boldsymbol{w}}_2 - \widehat{\boldsymbol{u}}_2; \lambda_3/\lambda_1, \gamma) \\ \vdots \\ \boldsymbol{\Theta}(\widehat{\boldsymbol{w}}_T - \widehat{\boldsymbol{u}}_T; \lambda_3/\lambda_1, \gamma) \end{pmatrix}, \tag{5.17}$$

$$(\widehat{w}_{m0}, \widehat{\boldsymbol{w}}_m) = \underset{w_{m0}, \boldsymbol{w}_m}{\arg\min} \left\{ \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2} \|\boldsymbol{w}_m - \widehat{\boldsymbol{u}}_m - \widehat{\boldsymbol{o}}_m\|_2^2 \right\}, \quad m = 1, \ldots, T, \tag{5.18}$$

$$\widehat{U} = \underset{U}{\arg\min} \left\{ \frac{\lambda_1}{2} \|\mathrm{vec}(\widehat{W}) - \mathrm{vec}(U) - \mathrm{vec}(\widehat{O})\|_2^2 + \lambda_2 \|D_r \mathrm{vec}(U)\|_{2,1} \right\}. \tag{5.19}$$

From the first-order condition of the minimization problems (5.18) and (5.19), we obtain

$$\frac{\partial}{\partial \boldsymbol{w}_m'} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m)|_{\boldsymbol{w}_m' = \widehat{\boldsymbol{w}}_m'} + \lambda_1 \begin{pmatrix} 0 \\ \widehat{\boldsymbol{w}}_m - \widehat{\boldsymbol{u}}_m - \widehat{\boldsymbol{o}}_m \end{pmatrix} = \mathbf{0},$$

$$-\lambda_1 \{\mathrm{vec}(\widehat{W}) - \mathrm{vec}(\widehat{U}) - \mathrm{vec}(\widehat{O})\} + \lambda_2 \partial_{\mathrm{vec}(U)} \|D_r \mathrm{vec}(U)\|_{2,1}|_{U=\widehat{U}} \ni \mathbf{0}.$$

Some algebra for them and (5.17) conclude Proposition 2.

□

Proof of Proposition 3. Since a limit point $\{\widehat{\boldsymbol{w}}_0^{(*)}, \widehat{W}^{(*)}, \widehat{U}^{(*)}, \widehat{O}^{(*)}, \widehat{B}^{(*)}, \widehat{S}^{(*)}\}$ satisfies the optimality condition of augmented Lagrangian, we have

$$-\lambda_1\{\widehat{W}^{(*)} - \widehat{U}^{(*)} - \widehat{O}^{(*)}\} + \widehat{S}^{(*)\top} A_{\mathcal{E}} = 0.$$

Then, similar to the proof of Proposition 2, it holds that

$$-\lambda_1(\psi(\widehat{\boldsymbol{w}}_1^{(*)} - \widehat{\boldsymbol{u}}_1^{(*)}; \lambda_3/\lambda_1, \gamma), \ldots, \psi(\widehat{\boldsymbol{w}}_T^{(*)} - \widehat{\boldsymbol{u}}_T^{(*)}; \lambda_3/\lambda_1, \gamma))^\top + \widehat{S}^{(*)\top} A_{\mathcal{E}} = 0.$$

This is one of the optimality conditions of the minimization problem concerning augmented Lagrangian (5.16). Moreover, other optimality conditions are the same as the minimization problem concerning the augmented Lagrangian (5.15). This concludes the Proposition 3. □

Proof of the Theorem 1. This proof is based on the similar steps of Wang et al. (2019) and Fan and Yin (2024). Although the modified ADMM does not explicitly require updates of B; the algorithm updates the value of U and S by the implicitly updated B in its construction (3.4). Specifically, B is updated to the value that exactly minimizes the function at each update of U in the inner loop of the gradient method. Then, the modified ADMM is equivalent to the following updates of ADMM:

$$\begin{aligned}
(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}) &= \arg\min_{\boldsymbol{w}_0, W} L_\nu(\boldsymbol{w}_0, W, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}), \\
(U^{(t+1)}, B^{(t+1)}) &= \arg\min_{U, B} L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U, O^{(t)}, B, S^{(t)}), \\
O^{(t+1)} &= \arg\min_{O} L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B^{(t+1)}, S^{(t)}), \\
S^{(t+1)} &= S^{(t)} + \nu(B^{(t+1)} - A_{\mathcal{E}} U^{(t+1)}).
\end{aligned}$$

Thus, it suffices to prove the theorem regarding the ADMM based on the above updates.

For the simplicity of the notation, we denote the object function as

$$Q(\boldsymbol{w}_0, W, U, O, B) = \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2}\|W - U - O\|_F^2 + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2}\|\boldsymbol{b}_{(m_1,m_2)}\|_2 + \sum_{m=1}^{T} P(\boldsymbol{o}_m, \lambda_3, \gamma),$$

and the partial of the object function as

$$H(\boldsymbol{w}_0, W, U, O) = \sum_{m=1}^{T} \frac{1}{n_m} L(w_{m0}, \boldsymbol{w}_m) + \frac{\lambda_1}{2}\|W - U - O\|_F^2 + \sum_{m=1}^{T} P(\boldsymbol{o}_m; \lambda_3, \gamma).$$

To prove the theorem, we show the following four steps:

1. $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)})$ is lower bounded for all $t \in \mathbb{N}$.
2. $\left\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\right\}$ is bounded.
3. $\left\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\right\}$ converges to a limit point, that is, $\lim_{t\to\infty} \left\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\right\} \to \left\{\boldsymbol{w}_0^{(*)}, W^{(*)}, U^{(*)}, O^{(*)}, B^{(*)}, S^{(*)}\right\}$.
4. Any limit point is a stationary point, that is, $\partial L_\nu(\boldsymbol{w}_0^{(*)}, W^{(*)}, U^{(*)}, O^{(*)}, B^{(*)}, S^{(*)}) \ni \mathbf{0}$.

Note that the objective function $Q(\boldsymbol{w}_0, W, U, O, B)$ is coercive over the feasible set, that is, $Q(\boldsymbol{w}_0, W, U, O, B) \to \infty$ if $A_\mathcal{E}U - B = 0$ and $\|\boldsymbol{w}_0, \mathrm{vec}(W), \mathrm{vec}(U), \mathrm{vec}(O), \mathrm{vec}(B)\|_2 \to \infty$. Moreover, as $\mathrm{Im}(A_\mathcal{E}) \subseteq \mathrm{Im}(I_{|\mathcal{E}|})$ with $\mathrm{Im}(\cdot)$ being the image of a matrix, there exists B' such that $A_\mathcal{E}U^{(t)} - B' = 0$. Therefore, we have

$$Q(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B') \geq \min_{\boldsymbol{w}_0, W, U, O, B} \{Q(\boldsymbol{w}_0, W, U, O, B) : A_\mathcal{E}U - B = 0\} > -\infty. \tag{5.20}$$

By the optimality condition for the updates of $(U^{(t+1)}, B^{(t+1)})$, it holds that for a sub-gradient $\boldsymbol{d}_{(m_1,m_2)}^{(t+1)} \in \partial(\lambda_2 r_{m_1,m_2}\|\boldsymbol{b}_{(m_1,m_2)}^{(t+1)}\|_2)$:

$$\boldsymbol{d}_{(m_1,m_2)}^{(t+1)} + \boldsymbol{s}_{(m_1,m_2)}^{(t)} + \nu(\boldsymbol{b}_{(m_1,m_2)}^{(t+1)} - (\boldsymbol{u}_{m_1}^{(t+1)} - \boldsymbol{u}_{m_2}^{(t+1)})) = \mathbf{0}, \quad \text{for } (m_1, m_2) \in \mathcal{E},$$
$$-\lambda_1(W^{(t+1)} - U^{(t+1)} - O^{(t)}) + A_\mathcal{E}^\top(S^{(t)} + \nu(B^{(t+1)} - A_\mathcal{E}U^{(t+1)})) = 0.$$

In addition, by the update of $S^{(t+1)}$, we have

$$\boldsymbol{s}^{(t+1)}_{(m_1,m_2)} = -\boldsymbol{d}^{(t+1)}_{(m_1,m_2)}, \quad \text{for } (m_1, m_2) \in \mathcal{E}.$$

For the subgradients $\boldsymbol{d}^{(t)}_{(m_1,m_2)} \in \partial(\lambda_2 r_{m_1,m_2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)}\|_2)$ and $\boldsymbol{d}'_{(m_1,m_2)} \in \partial(\lambda_2 r_{m_1,m_2} \|\boldsymbol{b}'_{(m_1,m_2)}\|_2)$ from the convexity, we have

$$\lambda_2 r_{m_1,m_2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)}\|_2 - \lambda_2 r_{m_1,m_2} \|\boldsymbol{b}'_{(m_1,m_2)}\|_2 \geq \langle \boldsymbol{d}^{(\prime)}_{(m_1,m_2)}, \boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)} \rangle$$

$$\begin{aligned}\lambda_2 r_{m_1,m_2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)}\|_2 + \langle \boldsymbol{d}^{(t)}_{(m_1,m_2)}, \boldsymbol{b}'_{(m_1,m_2)} - \boldsymbol{b}^{(t)}_{(m_1,m_2)} \rangle &\geq \lambda_2 r_{m_1,m_2} \|\boldsymbol{b}'_{(m_1,m_2)}\|_2 \\ &\quad + \langle \boldsymbol{d}'_{(m_1,m_2)} - \boldsymbol{d}^{(t)}_{(m_1,m_2)}, \boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)} \rangle.\end{aligned}$$

Then, we have

$$\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) \\
&= Q(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}) \\
&+ \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \boldsymbol{s}^{(t)\top}_{(m_1,m_2)} (\boldsymbol{b}^{(t)}_{(m_1,m_2)} - (\boldsymbol{u}^{(t)}_{m_1} - \boldsymbol{u}^{(t)}_{m_2})) + \frac{\nu}{2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)} - (\boldsymbol{u}^{(t)}_{m_1} - \boldsymbol{u}^{(t)}_{m_2})\|_2^2 \right\} \\
&= H(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}) + \sum_{(m_1,m_2)\in\mathcal{E}} r_{m_1,m_2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)}\|_2 + \\
&+ \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \boldsymbol{d}^{(t)\top}_{(m_1,m_2)} (\boldsymbol{b}'_{(m_1,m_2)} - \boldsymbol{b}^{(t)}_{(m_1,m_2)}) + \frac{\nu}{2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)}\|_2^2 \right\} \\
&\geq H(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}) + \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ r_{m_1,m_2} \|\boldsymbol{b}'_{(m_1,m_2)}\|_2 + \langle \boldsymbol{d}'_{(m_1,m_2)} - \boldsymbol{d}^{(t)}_{(m_1,m_2)}, \boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)} \rangle \right. \\
&\left. + \frac{\nu}{2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)}\|_2^2 \right\} \\
&\geq Q(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B') \\
&+ \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \frac{\nu}{2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)}\|_2^2 - 2\lambda_2 r_{m_1,m_2} \|\boldsymbol{b}^{(t)}_{(m_1,m_2)} - \boldsymbol{b}'_{(m_1,m_2)}\|_2 \right\} \\
&> -\infty
\end{aligned}$$

The second inequality is based on the fact $\|\partial_{\boldsymbol{b}} \|\boldsymbol{b}\|_2\|_2 \leq 1$ for $\forall \boldsymbol{b} \in \mathbb{R}^p$ and Cauchy – Schwarz inequality. The last inequality is derived from the (5.20). This completes the proof of the first step.

To bound $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t+1)})$, we show the following bounds regarding each update:

(i) $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)})$,

(ii) $L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B^{(t+1)}, S^{(t)})$,

(iii) $L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B^{(t+1)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t)})$,

(iv) $L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t+1)})$.

For (i), we have

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) \\
&= \sum_{m=1}^{T} \frac{1}{n_m} \left\{ L(w_{m0}^{(t)}, \boldsymbol{w}_m^{(t)}) - L(w_{m0}^{(t+1)}, \boldsymbol{w}_m^{(t+1)}) \right\} \\
&+ \frac{\lambda_1}{2} \left\{ \|W^{(t)} - U^{(t)} - O^{(t)}\|_F^2 - \|W^{(t+1)} - U^{(t)} - O^{(t)}\|_F^2 \right\} \\
&\geq \frac{\lambda_1}{2} \|W^{(t)} - W^{(t+1)}\|_F^2.
\end{aligned}
$$

The inequality is derived from the convexity of the loss function. For (ii), we have

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B^{(t+1)}, S^{(t)}) \\
&= \frac{\lambda_1}{2} \left\{ \|W^{(t+1)} - U^{(t)} - O^{(t)}\|_F^2 - \|W^{(t+1)} - U^{(t+1)} - O^{(t)}\|_F^2 \right\} \\
&+ \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \|\boldsymbol{b}_{(m_1,m_2)}^{(t)}\|_2 - \|\boldsymbol{b}_{(m_1,m_2)}^{(t+1)}\|_2 \right\} \\
&+ \langle S^{(t)}, B^{(t)} - A_\mathcal{E} U^{(t)} - (B^{(t+1)} - A_\mathcal{E} U^{(t+1)}) \rangle \\
&+ \frac{\nu}{2} \left\{ \|B^{(t)} - A_\mathcal{E} U^{(t)}\|_F^2 - \|B^{(t+1)} - A_\mathcal{E} U^{(t+1)}\|_F^2 \right\} \\
&= \frac{\lambda_1}{2} \|U^{(t)} - U^{(t+1)}\|_F^2 + \lambda_2 \sum_{(m_1,m_2)\in\mathcal{E}} \left\{ \|\boldsymbol{b}_{(m_1,m_2)}^{(t)}\|_2 - \|\boldsymbol{b}_{(m_1,m_2)}^{(t+1)}\|_2 \right\} \\
&+ \nu \langle B^{(t+1)} - A_\mathcal{E} U^{(t+1)}, A_\mathcal{E}(U^{(t)} - U^{(t+1)}) \rangle \\
&+ \langle D^{(t+1)}, B^{(t+1)} - B^{(t)} \rangle + \nu \langle A_\mathcal{E} U^{(t)} - B^{(t+1)}, B^{(t)} - B^{(t+1)} \rangle \\
&+ \frac{\nu}{2} \left\{ \|B^{(t)} - A_\mathcal{E} U^{(t)}\|_F^2 - \|B^{(t+1)} - A_\mathcal{E} U^{(t)}\|_F^2 \right\} \\
&\geq \frac{\lambda_1}{2} \|U^{(t)} - U^{(t+1)}\|_F^2 + \frac{\nu}{2} \|(B^{(t)} - A_\mathcal{E} U^{(t)}) - (B^{(t+1)} - A_\mathcal{E} U^{(t+1)})\|_F^2
\end{aligned}
$$

where $D^{(t+1)}$ is a $|\mathcal{E}| \times p$ matrix whose each row is $\boldsymbol{d}^{(t+1)}_{(m_1,m_2)}$. The second equality is derived from the optimality condition of $(U^{(t+1)}, B^{(t+1)})$, and the cosine rule $\|A - B\|_F^2 - \|A - C\|_F^2 + 2\langle A - C, B - C\rangle = \|B - C\|_F^2$. The inequality is derived from the convexity of $\|\boldsymbol{b}^{(t+1)}_{(m_1,m_2)}\|_2$. For (iii), we have

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t)}, B^{(t+1)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t)}) \\
&= \frac{\lambda_1}{2}\left\{\|W^{(t+1)} - U^{(t+1)} - O^{(t)}\|_F^2 - \|W^{(t+1)} - U^{(t+1)} - O^{(t+1)}\|_F^2\right\} \\
&+ \sum_{m=1}^{T}\left\{P(\boldsymbol{o}_m^{(t)}, \lambda_3, \gamma) - P(\boldsymbol{o}_m^{(t+1)}, \lambda_3, \gamma)\right\} \\
&\geq \frac{\lambda_1 - C_p}{2}\|O^{(t+1)} - O^{(t)}\|_F^2.
\end{aligned}
$$

The inequality is derived from the weakly convexity of the regularization term with the constant $C_p > 0$. For (iv), we have

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t+1)}) \\
&= \langle S^{(t)} - S^{(t+1)}, B^{(t+1)} - A_{\mathcal{E}} U^{(t+1)}\rangle \\
&= -\frac{1}{\nu}\|S^{(t)} - S^{(t+1)}\|_F^2 \\
&\geq -\frac{1}{\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})}\|A_{\mathcal{E}}^\top(S^{(t)} - S^{(t+1)})\|_F^2 \\
&= -\frac{\lambda_1^2}{\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})}\|W^{(t)} - U^{(t)} - O^{(t-1)} - (W^{(t+1)} - U^{(t+1)} - O^{(t)})\|_F^2 \\
&\geq -\frac{\lambda_1^2}{\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})}\left\{\|W^{(t)} - W^{(t+1)}\|_F^2 + \|U^{(t)} - U^{(t+1)}\|_F^2 + \|O^{(t-1)} - O^{(t)}\|_F^2\right\},
\end{aligned}
$$

By combining the above upper bound of each update, we obtain

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) - L_\nu(\boldsymbol{w}_0^{(t+1)}, W^{(t+1)}, U^{(t+1)}, O^{(t+1)}, B^{(t+1)}, S^{(t+1)}) \\
&\geq \lambda_1\left(\frac{\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}}) - 2\lambda_1}{2\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})}\right)\left(\|W^{(t)} - W^{(t+1)}\|_F^2 + \|U^{(t)} - U^{(t+1)}\|_F^2\right) \\
&+ \frac{\nu}{2}\left\|(B^{(t)} - A_{\mathcal{E}}U^{(t)}) - (B^{(t+1)} - A_{\mathcal{E}}U^{(t+1)})\right\|_F^2 \\
&+ \frac{\lambda_1 - C_p}{2}\|O^{(t)} - O^{(t+1)}\|_F^2 - \frac{\lambda_1^2}{\nu\lambda_{++}(A_{\mathcal{E}}^\top A_{\mathcal{E}})}\|O^{(t-1)} - O^{(t)}\|_F^2.
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
&L_\nu(\boldsymbol{w}_0^{(0)}, W^{(0)}, U^{(0)}, O^{(0)}, B^{(0)}, S^{(0)}) - L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) \\
&\geq \sum_{l=0}^{t} \left\{ \lambda_1 \left(\frac{\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E}) - 2\lambda_1}{2\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})} \right) \left(\|W^{(l)} - W^{(l+1)}\|_F^2 + \|U^{(l)} - U^{(l+1)}\|_F^2 \right) \right. \\
&\left. + \frac{\nu}{2} \left\| (B^{(t)} - A_\mathcal{E} U^{(t)}) - (B^{(t+1)} - A_\mathcal{E} U^{(t+1)}) \right\|_F^2 \right\} \qquad (5.21) \\
&+ \left(\frac{\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})(\lambda_1 - C_p) - 2\lambda_1^2}{2\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})} \right) \sum_{l=1}^{t} \|O^{(l)} - O^{(l-1)}\|_F^2 \\
&+ \frac{\lambda_1 - C_p}{2} \|O^{(t)} - O^{(t+1)}\|_F^2 - \frac{\lambda_1^2}{\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})} \|O^{(0)}\|_F^2.
\end{aligned}
$$

From this, $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)})$ is upper bounded by $L_\nu(\boldsymbol{w}_0^{(0)}, W^{(0)}, U^{(0)}, O^{(0)}, B^{(0)}, S^{(0)}) + \frac{\lambda_1^2}{\nu\lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})} \|O^{(0)}\|_F^2$. Thus, $Q(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B')$ and $\|B^{(t)} - A_\mathcal{E} U^{(t)}\|_F^2$ are also upper bounded. Since the objective function $Q(\boldsymbol{w}_0, W, U, O, B)$ is coercive over the feasible set, $\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}\}$ is bounded. Then, from the following inequality:

$$
\|S^{(t)}\|_F^2 \leq \lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})^{-1} \|A_\mathcal{E}^\top S^{(t)}\|_F^2 = \lambda_{++}(A_\mathcal{E}^\top A_\mathcal{E})^{-1} \lambda_1^2 \|W^{(t+1)} - U^{(t+1)} - O^{(t)}\|_F^2, \qquad (5.22)
$$

$S^{(t)}$ is also bounded. Furthermore, from the lower boundness of the $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)})$, the value of right hand side of (5.21) converges to some non-negative constant without the last term as $t \to \infty$. Therefore, we obtain $\lim_{t\to\infty} \|W^{(t)} - W^{(t+1)}\|_F^2 = 0$, $\lim_{t\to\infty} \|U^{(t)} - U^{(t+1)}\|_F^2 = 0$, $\lim_{t\to\infty} \|(B^{(t)} - A_\mathcal{E} U^{(t)}) - (B^{(t+1)} - A_\mathcal{E} U^{(t+1)})\|_F^2 = 0$, and $\lim_{t\to\infty} \|O^{(t)} - O^{(t+1)}\|_F^2 = 0$. Moreover, from triangle inequality

$$
\|B^{(t)} - B^{(t+1)}\|_F^2 \leq \|(B^{(t)} - A_\mathcal{E} U^{(t)}) - (B^{(t+1)} - A_\mathcal{E} U^{(t+1)})\|_F^2 + \|A_\mathcal{E}(U^{(t)} - U^{(t+1)})\|_F^2
$$

and the fact $\|A_\mathcal{E}(U^{(t)} - U^{(t+1)})\| \leq \lambda_+(A_\mathcal{E}^\top A_\mathcal{E}) \|U^{(t)} - U^{(t+1)}\|_F^2$, we obtain $\lim_{t\to\infty} \|B^{(t)} - B^{(t+1)}\|_F^2 = 0$. These convergence and (5.22) also imply $\lim_{t\to\infty} \|S^{(t)} - S^{(t+1)}\|_F^2 = 0$. Since the estimation of $\boldsymbol{w}_0$ calculated by Algorithm 3 only depends on $U^{(t-1)}$ and $O^{(t-1)}$, $\lim_{t\to\infty} \|\boldsymbol{w}_0^{(t)} - \boldsymbol{w}_0^{(t+1)}\|_2^2 = 0$. This concludes the steps 2 and 3.

To prove the limit point of the ADMM algorithm satisfies the optimality condition, we show

$$\lim_{t\to\infty} \left\| \partial L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}) \right\|_F = 0.$$

We denote $L_\nu(\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}) = L_\nu^{(t)}$. From the optimality condition of update for $\boldsymbol{o}_m^{(t)}$, we have $\partial_{\boldsymbol{o}_m} L_\nu^{(t)} \ni \mathbf{0}$. We have

$$\begin{aligned}
\lim_{t\to\infty} \left\| \frac{\partial L_\nu^{(t)}}{\partial(\boldsymbol{w}_0, W)} \right\|_F &\le \lim_{t\to\infty} \sum_{m=1}^{T} \left\| \frac{\partial L_\nu^{(t)}}{\partial \boldsymbol{w}_m'} \right\|_2 \\
&= \lim_{t\to\infty} \sum_{m=1}^{T} \left\| \lambda_1(\boldsymbol{w}_m^{(t-1)} - \boldsymbol{u}_m^{(t-1)} - \boldsymbol{o}_m^{(t-1)}) - \lambda_1(\boldsymbol{w}_m^{(t)} - \boldsymbol{u}_m^{(t)} - \boldsymbol{o}_m^{(t)}) \right\|_2 = 0
\end{aligned}$$

and

$$\begin{aligned}
\lim_{t\to\infty} \|\partial_B L_\nu^{(t)}\|_F &\le \lim_{t\to\infty} \sum_{(m_1,m_2)\in\mathcal{E}} \left\| \partial_{\boldsymbol{b}_{(m_1,m_2)}} L_\nu^{(t)} \right\|_2 \\
&= \lim_{t\to\infty} \sum_{(m_1,m_2)\in\mathcal{E}} \left\| \boldsymbol{d}_{(m_1,m_2)}^{(t)} + \boldsymbol{s}_{(m_1,m_2)}^{(t)} + \nu(\boldsymbol{b}_{(m_1,m_2)}^{(t)} - (\boldsymbol{u}_{m_1}^{(t)} - \boldsymbol{u}_{m_2}^{(t)})) \right\|_2 \\
&= \lim_{t\to\infty} \sum_{(m_1,m_2)\in\mathcal{E}} \nu \left\| \boldsymbol{b}_{(m_1,m_2)}^{(t)} - (\boldsymbol{u}_{m_1}^{(t)} - \boldsymbol{u}_{m_2}^{(t)}) \right\|_2 \\
&\le \lim_{t\to\infty} \nu \left\| B^{(t)} - A_{\mathcal{E}} U^{(t)} \right\|_F^2 = 0.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\lim_{t\to\infty} \left\| \frac{\partial L_\nu^{(t)}}{\partial U} \right\|_F &= \lim_{t\to\infty} \left\| -\lambda_1(W^{(t)} - U^{(t)} - O^{(t)}) - \langle A_{\mathcal{E}}, S^{(t)} + \nu(B^{(t)} - A_E U^{(t)}) \rangle \right\|_F \\
&= \lim_{t\to\infty} \Big\| \lambda_1(W^{(t)} - W^{(t+1)} - (U^{(t)} - U^{(t+1)}) \\
&\quad + \nu \langle A_{\mathcal{E}}, (B^{(t)} - B^{(t+1)}) - A_{\mathcal{E}}(U^{(t)} - U^{(t+1)}) \rangle \Big\|_F \\
&\le \lim_{t\to\infty} \lambda_1 \left\{ \|W^{(t)} - W^{(t+1)}\|_F + \|U^{(t)} - U^{(t+1)}\|_F \right\} \\
&\quad + \nu \left\{ \|A_{\mathcal{E}}^\top (B^{(t)} - B^{(t+1)})\|_F + \|A_{\mathcal{E}}^\top A_{\mathcal{E}} (U^{(t)} - U^{(t+1)})\|_F \right\} = 0.
\end{aligned}$$

The last equality is derived from the fact $\left\| A_{\mathcal{E}}^\top (B^{(t)} - B^{(t+1)}) \right\|_F^2 \le \lambda_+(A_{\mathcal{E}}^\top A_{\mathcal{E}}) \left\| B^{(t)} - B^{(t+1)} \right\|_F^2$

and $\left\|A_{\mathcal{E}}^{\top}A_{\mathcal{E}}(U^{(t)}-U^{(t+1)})\right\|_F^2 \leq \lambda_+(A_{\mathcal{E}}^{\top}A_{\mathcal{E}})\left\|A_{\mathcal{E}}(U^{(t)}-U^{(t+1)})\right\|_F^2$. Finally, we have

$$\begin{aligned}\lim_{t\to\infty}\left\|\frac{\partial L_\nu^{(t)}}{\partial S}\right\|_F &= \lim_{t\to\infty}\left\|B^{(t)}-A_{\mathcal{E}}U^{(t)}\right\|_F \\ &= \lim_{t\to\infty}\frac{1}{\nu}\left\|S^{(t+1)}-S^{(t)}\right\|_F = 0.\end{aligned}$$

Therefore, we have $\lim_{t\to\infty}\left\|\partial L_\nu^{(t)}\right\|_F = 0$.

Because $\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\}$ is bounded and converges to a limit point denoted by $\{\boldsymbol{w}_0^{(*)}, W^{(*)}, U^{(*)}, O^{(*)}, B^{(*)}, S^{(*)}\}$. Moreover, it holds that $\lim_{t\to\infty}\|\partial L_\nu^{(t)}\|_F = \|\partial L_\nu^{(*)}\|_F = 0$. From the definition of general subgradient, we have $\partial L_\nu^{(*)} \ni \mathbf{0}$. Therefore, the sequence $\{\boldsymbol{w}_0^{(t)}, W^{(t)}, U^{(t)}, O^{(t)}, B^{(t)}, S^{(t)}\}$ has at least a limit point $\{\boldsymbol{w}_0^{(*)}, W^{(*)}, U^{(*)}, O^{(*)}, B^{(*)}, S^{(*)}\}$ and any limit point is a stationary point. This completes the proof. □

Chapter 6

Concluding remarks

6.1 Summary

In this thesis, we proposed two novel multi-task learning methods that consider the clustering of tasks.

The first one is Multi-Task Learning via ConVeX clustering (MTLCVX). Because the parameters are split into those for regression and clustering, we can expect to reduce the shrinkages between irrelevant tasks caused by fused group regularization. In simulation studies, our proposed methods showed better results than the existing method based on the network lasso called MTLNL in almost all cases. MTLCVX can be more robust against noise in the weights than MTLNL. For the application to real data, if there are multiple clusters in the data, MTLCVX showed better performance.

Second, we proposed a robust multi-task learning method called Multi-Task Learning via Robust Regularized Clustering (MTLRRC). To perform the clustering of tasks and detection of outlier tasks simultaneously, we incorporated regularization terms based on robust regularized clustering (RRC), which can detect outlier samples by selecting outlier parameters through group sparse penalties. We showed that the solution of the RRC obtained by the BCD algorithm shares the same optimality condition with

convex clustering whose loss function is replaced by the multivariate robust loss function. Thus, MTLRRC is expected to perform robust clustering of tasks. Furthermore, the solution of MTLRRC by the BCD algorithm is also viewed as a convergence point of alternative optimization that involves the RRC for tasks and regression problems, reducing the shrinkage of outlier tasks toward the estimated centroid. To mitigate computational costs, we developed an estimation algorithm based on the modified ADMM and provided the theoretical convergence guarantees. Numerical studies showed that if there are outlier tasks with large unique characteristics, MTLRRC effectively detects them and improves estimation and prediction accuracy. Moreover, the application to the three real datasets showed that they have three different characteristics. Specifically, one has two clusters with potentially outlier tasks, one has one cluster with outlier tasks with high probability, and one has one cluster without outlier tasks. Interestingly, MTLRRC with group SCAD performed the best among the competing methods regardless of the existence of cluster structure and outlier tasks.

6.2 Limitations and future works

Finally, we discuss the limitations of our proposed methods and future works.

First, our proposed methods only consider the situations where the obtained features are identical across the tasks. Because the tasks may be obtained from heterogeneous data sources and experiment environments in practical situations, some of the features may differ across the tasks. Thus, it is desirable to develop MTL methods that consider feature heterogeneity. An approach is to map the features to a common feature space. It is also interesting to develop MTL methods that estimate feature mapping and cluster tasks in the mapped space simultaneously.

Second, we do not consider the sparsity of the variables. Although sparse approach methods consider the joint sparsity across the tasks, having different sparse patterns

among different clusters would be favorable. However, to cluster the variables that exist in different spaces may be non-trivial. For instance, if a regression coefficient is zero for a task and non-zero for another task, the distance between tasks would be large depending on only one variable regardless of other common non-zero features. Therefore, using L_1 penalty to shrink a difference would be reasonable in sparse settings. In particular, Tang et al. (2021) proposed the regularization term that shrinks a regression coefficient towards either zero or the cluster center, whichever is closer. Although the method can prevent the shrinkage of nonzero variables toward zero, the way of estimating the cluster center itself is not given. In other words, it would be essential to consider appropriate clustering methods for cluster centers with different nonzero variables.

Third, the selection of tuning parameters is also challenging in light of computational complexity. In MTL, the total dimension of variables tends to be huge, as the number of tasks increases. The validation for selecting tuning parameters takes much computation time compared to single-task learning. Thus, developing a method that can effectively select the tuning parameters, such as information criteria, is necessary. One of the approaches would be to reformulate the MTL methods as those in the Bayesian framework. In the Bayesian perspective, the tuning parameters are determined by maximizing marginal likelihood, which can be computed by algorithms such as the EM algorithm. In addition, the extension to a Bayesian model enables us to analyze estimation and prediction uncertainty by calculating a posterior distribution. Consequently, the extension to the Bayesian framework is attractive in some aspects. However, calculating the posterior distribution itself is computationally demanding. Establishing an efficient calculation method would also be challenging in model selection and Bayesian multi-task learning.

Many challenges remain in the field of multi-task learning. We leave these topics as future work.

Bibliography

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.

Antoniadis, A. (2007). Wavelet methods in statistics: Some recent developments and their applications. *Statistics Surveys*, **1**, 16–55.

Argyriou, A., Evgeniou, T., and Pontil, M. (2006). Multi-task feature learning. *Advances in Neural Information Processing Systems*, **19**, 41–48.

Argyriou, A., Pontil, M., Ying, Y., and Micchelli, C. A. (2007). A spectral regularization framework for multi-task structure learning. *Advances in Neural Information Processing Systems*, **20**, 25–32.

Bakker, B. and Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, **4**, 83–99.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, **3**(1), 1–122.

Caruana, R. (1997). Multitask learning. *Machine Learning*, **28**, 41–75.

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.

Chi, E. C. and Lange, K. (2015). Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, **24**(4), 994–1013.

Cook, R. D., Li, B., and Chiaromonte, F. (2010). Envelope models for parsimonious and efficient multivariate linear regression. *Statistica Sinica*, **20**(3), 927–960.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. (2018). Learning to learn around a common mean. *Advances in Neural Information Processing Systems*, 10190–10200.

Deng, D., Shahabi, C., Demiryurek, U., and Zhu, L. (2017). Situation aware multi-task learning for traffic prediction. In *2017 IEEE International Conference on Data Mining*, 81–90.

Dondelinger, F., Mukherjee, S., and Alzheimer's Disease Neuroimaging Initiative. (2020). The joint lasso: high-dimensional regression for group structured data. *Biostatistics*, **21**(2), 219–235.

Evgeniou, T., Micchelli, C. A., and Pontil, M. (2005). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, **6**, 615–637.

Fan, J., Gao, Y., and Luo, H. (2008). Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *IEEE Transactions on Image Processing*, **17**(3), 407–426.

Fan, Y. and Yin, G. (2024). Gaussian mixture models with concave penalized fusion. *Statistica Sinica*, **3**, 2115–2139.

Gannaz, I. (2007). Robust estimation and wavelet thresholding in partially linear models. *Statistics and Computing*, **17**, 293–310.

Gong, P., Ye, J., and Zhang, C. (2012). Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 895–903.

Hallac, D., Leskovec, J., and Boyd, S. (2015). Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–396.

Hampel, F. R. (1985). The breakdown points of the mean combined with some rejection rules. *Technometrics*, **27**(2), 95–107.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

Han, L. and Zhang, Y. (2015). Learning multi-level task groups in multi-task learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, **29**(1), 2638–2644.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity*. Chapman and Hall/CRC.

He, X., Alesiani, F., and Shaker, A. (2019). Efficient and scalable multi-task regression on massive number of tasks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, **462**, 3763–3770.

Hocking, T. D., Joulin, A., Bach, F., and Vert, J. P. (2011). Clusterpath an algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th International Conference on Machine Learning*, 745–752.

Huang, J., Breheny, P., and Ma, S. (2012). A selective review of group selection in high-dimensional models. *Statistical Science*, **27**(4), 481–499.

Jalali, A., Sanghavi, S., Ruan, C., and Ravikumar, P. (2010). A dirty model for multi-task learning. *Advances in Neural Information Processing Systems*, **23**, 964–972.

Kang, Z., Grauman, K., and Sha, F. (2011). Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, 521–528.

Katayama, S. and Fujisawa, H. (2017). Sparse and robust linear regression: An optimization algorithm and its statistical properties. *Statistica Sinica*, **27**(3), 1243–1264.

Lange, K. (2016). *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics Philadelphia.

Lee, M. and Su, Z. (2020). A review of envelope models. *International Statistical Review*, **88**(3), 658–676.

Li, L., He, X., and Borgwardt, K. (2018). Multi-target drug repositioning by bipartite block-wise sparse multi-task learning. *BMC Systems Biology*, **12**(4), 85–97.

Lindsten, F., Ohlsson, H., and Ljung, L. (2011). Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop*, 201–204.

Liu, H., Palatucci, M., and Zhang, J. (2009). Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th International Conference on Machine Learning*, 649–656.

Lounici, K., Pontil, M., Tsybakov, A. B., and Geer, S.,v. d. (2009). Taking advantage of sparsity in multi-task learning. Preprint, arXiv:0903.1468.

Lounici, K., Pontil, M., Geer, S.,v. d, and Tsybakov, A. B. (2011). Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics*, **39**(4), 2164–2204.

Maronna, R. A. (1976). Robust m-estimators of multivariate location and scatter. *Annals of Statistics*, **4**(1), 51–67.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate o(1/k^2). *Proceedings of the USSR Academy of Sciences*, **269**, 543–547.

Ngai, H. V., Luc, D. T., and Théra, M. (2000). Approximate convex functions. *Journal of Nonlinear and Convex Analysis*, **1**(2), 155–176.

Obozinski, G., Taskar, B., and Jordan, M. I. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, **20**, 231–252.

Parameswaran, S. and Weinberger, K. Q. (2010). Large margin multi-task metric learning. *Advances in Neural Information Processing Systems*, **23**, 1867–1875.

Pelckmans, K., De Brabanter, J., Suykens, J., and De Moor, B. (2005). Convex clustering shrinkage. In *PASCAL workshop on Statistics and Optimization of Clustering workshop*, **1524**.

Pong, T. K., Tseng, P., Ji, S., and Ye, J. (2010). Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, **20**(6), 3465–3489.

Quan, Z. and Chen, S. (2020). Robust convex clustering. *Soft Computing*, **24**(2), 731–744.

She, Y. and Owen, A. B. (2011). Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, **106**(494), 626–639.

Shimamura, K. and Kawano, S. (2021). A bayesian approach to multi-task learning with network lasso. Preprint, arXiv:1402.6455.

Shimmura, R. and Suzuki, J. (2022). Converting admm to a proximal gradient for efficient sparse estimation. *Japanese Journal of Statistics and Data Science*, **5**, 725–745.

Sun, D., Toh, K.-C., and Yuan, Y. (2021). Convex clustering: Model, theoretical guarantee and efficient algorithm. *Journal of Machine Learning Research*, **22**(1), 427–458.

Tan, K. M. and Witten, D. (2015). Statistical properties of convex clustering. *Electronic Journal of Statistics*, **9**(2), 2324–2347.

Tang, X., Xue, F., and Qu, A. (2021). Individualized multidirectional variable selection. *Journal of the American Statistical Association*, **116**(535), 1280–1296.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B; Statistical Methodology*, **67**(1), 91–108.

Turlach, B. A., Venables, W. N., and Wright, S. J. (2005). Simultaneous variable selection. *Technometrics*, **47**(3), 349–363.

Wang, Y., Yin, W., and Zeng, J. (2019). Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, **78**, 29–63.

Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., Rohr, P.,v, Thiele, L., et al. (2004). Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biology*, **5**, 1–13.

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** (2), 35–63.

Yamada, M., Koh, T., Iwata, T., Shawe-Taylor, J., and Kaski, S. (2017). Localized lasso for high-dimensional regression. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, **54**, 325–333.

Yao, Y., Cao, J., and Chen, H. (2019). Robust task grouping with representative tasks for clustered multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1408–1417.

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

Zhang, X., Liu, J., and Zhu, Z. (2024). Learning coefficient heterogeneity over networks: A distributed spanning-tree-based fused-lasso regression. *Journal of the American Statistical Association*, **119**(545), 485–497.

Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, **32**(12), 5586–5609.

Zhong, W. and Kwok, J. T. Y. (2012). Convex multitask learning with flexible task clusters. In *Proceedings of the 29th International Conference on Machine Learning*, 483–490.

Zhou, J., Chen, J., and Ye, J. (2011). Clustered multi-task learning via alternating structure optimization. *Advances in Neural Information Processing Systems*, **24**, 702–710.

Zhou, J., Yuan, L., Liu, J., and Ye, J. (2011). A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 814–822.

Zhou, Q. and Zhao, Q. (2016). Flexible clustered multi-task learning by learning representative tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(2), 266–278.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**(476), 1418–1429.