Multi-Task Learning via Structured Regularization:

Formulations, Algorithms, and Applications

by

Jianhui Chen

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved August 2011 by the
Graduate Supervisory Committee:

Jieping Ye, Chair
Sudhir Kumar
Huan Liu
Guoliang Xue

ARIZONA STATE UNIVERSITY

August 2011

ABSTRACT

Multi-task learning (MTL) aims to improve the generalization performance (of the resulting classifiers) by learning multiple related tasks simultaneously. Specifically, MTL exploits the intrinsic task relatedness, based on which the informative domain knowledge from each task can be shared across multiple tasks and thus facilitate the individual task learning. It is particularly desirable to share the domain knowledge (among the tasks) when there are a number of related tasks but only limited training data is available for each task.

Modeling the relationship of multiple tasks is critical to the generalization performance of the MTL algorithms. In this dissertation, I propose a series of MTL approaches which assume that multiple tasks are intrinsically related via a shared low-dimensional feature space. The proposed MTL approaches are developed to deal with different scenarios and settings; they are respectively formulated as mathematical optimization problems of minimizing the empirical loss regularized by different structures. For all proposed MTL formulations, I develop the associated optimization algorithms to find their globally optimal solution efficiently. I also conduct theoretical analysis for certain MTL approaches by deriving the globally optimal solution recovery condition and the performance bound. To demonstrate the practical performance, I apply the proposed MTL approaches on different real-world applications: (1) Automated annotation of the Drosophila gene expression pattern images; (2) Categorization of the Yahoo web pages. Our experimental results demonstrate the efficiency and effectiveness of the proposed algorithms.

To My Dear Parents

ACKNOWLEDGEMENTS

This dissertation could not have been possible without support and encouragement from many people.

First and foremost, I want to extend my deepest gratitude to Professor Jieping Ye, my Ph.D. advisor and the best mentor I could ever have. In pursuit of my Ph.D, his helps are always around, from spiritual encouragement, daily life, research methodologies, technical writing, and presentation skill. What I have learned from Professor Ye are unique treasures for my future career. Having Professor Ye as my Ph.D. advisor is one of the most lucky things in my life. I also want to thank my dissertation committee members: Professor Sudhir Kumar, Professor Huan Liu, and Professor Guoliang Xue, for their guidance, support, and feedback.

It is a rewarding and pleasant experience to work as a student research associate in the Machine Learning Lab and the Center for Evolutionary Medicine and Informatics at Arizona State University. The lab and the center have been sources of friendships, good advice, as well as collaboration. Especially I would like to thank the following people for their great help and valuable interaction: Betul Ceran, Rita Chattopadhyay, Rashmi Dubey, Bernard Van Emden, Kristi Garboushian, Shuiwang Ji, Ji Liu, Jun Liu, Yashu Liu, Wayne Parkhurst, Rinkal Patel, Bao-Hong Shen, Liang Sun, Qian Sun, Ramesh Thulasiram, Carol Williams, Jason Wolf, Shuo Xiang, Lei Yuan, Sen Yang, and Jiayu Zhou. As incredible collaborators and greatest friends, they inspire me a lot through constructive discussion, interesting seminars, and excellent team work.

My time at Arizona State University was made enjoyable in another large part due to many of other friends. I would like to give very special thanks to Zhen Li; I also would like to thank other friends: the Andersons, Huiji Gao, Gregory L. Heileman, Yuheng Hu, Pramod A. Jamkhedkar, Nan Li, Youzuo Lin, Xufeng Liu, Wenbin Luo, Hualin Lv, Yan Qi, Lei Tang, Zhizhong Tang, Kai Tu, Shanshan Wang, Xufei Wang, Wu Li, Wei Xu, Liuxian Zhang, Sushu Zhang, Jicheng Zhao, and Yanan Zhao. Although we are diverse in nationality, age, research discipline, and occupation, we share lots of common interests and personalities. I will treasure our friendship forever.

Last but not least, I want to thank my parents for their invaluable love. They are always supporting me and encouraging me with their best wishes.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In many real-world pattern classification problems [1, 2], the tasks can often be divided into several (sub)tasks. For example, the problem of landmine detection [3] consists of the tasks of classifying the objects (represented by the radar images) from different landmine fields into either the landmine category or the clutter category; the problem of *Drosophila* gene expression pattern images annotation consists of the tasks of assigning the controlled vocabulary (CV) terms to the gene expression images groups. Traditionally multiple tasks are solved or learned via the *single-task learning* (STL) scheme; in STL, the learning (training) processes of multiple tasks are separate (one task is learned at a time), and the learned predictive model for each task is applied independently for generalization on the unseen data, as illustrated in the left plot of Figure 1.1. Commonly used STL algorithms include support vector machine, ridge regression, and logistical regression.

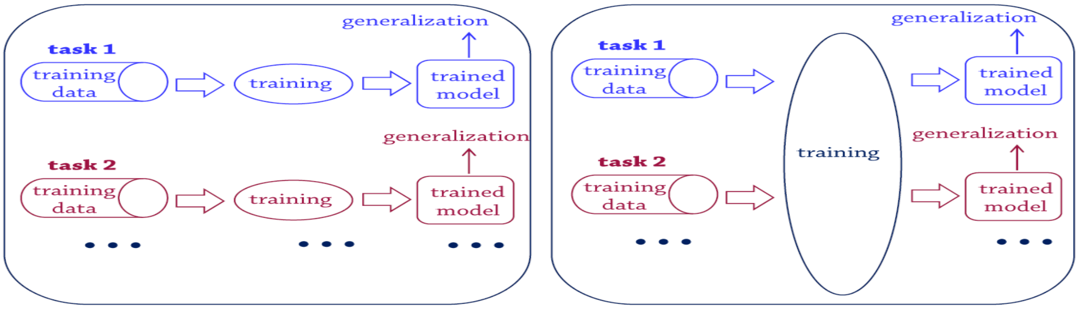


Figure 1.1: Illustration of the single task learning model (left) and the multi-task learning model (right).

In reality, the tasks are often related via certain underlying relationship. For example, the tasks of classifying the objects (using the radar images) from geographically different landmine fields are related, as the landmine is represented by some common low-level features on the radar images; the tasks of assigning the CV terms to gene expression images groups are related, as the images from the same group share certain anatomical and developmental structures. Simply learning the tasks separately may lead to suboptimal generalization performance. Therefore it is desirable to incorporate the underlying relation (among the tasks) into the algorithms for learning multiple (related) tasks. This corresponds to a general learning scheme called *multi-task learning* (MTL) [4].

MTL aims to improve the generalization performance of the classifiers by learning from multiple related tasks. It can be achieved by learning the tasks simultaneously and meanwhile exploiting the intrinsic relatedness among the tasks. Specifically, in MTL the learning process for all tasks are inter-related via the modeled task relatedness, and the learned predictive model for each task is then applied for generalization independently, as depicted in the right plot of Figure 1.1. Based on the MTL scheme, the informative domain knowledge of each task is allowed to be shared across the tasks, thus facilitating individual task learning. It is particularly desirable to share such knowledge across the tasks when there are a number of related tasks but only limited training data is available for each task.

This chapter is organized as follows; in Section 1.1, I discuss several representative algorithms for multi-task learning; in Section 1.2 I give a brief comparison among different learning methods; in Section 1.3 I present two applications of the proposed MTL algorithms; in Section 1.4 I summarize the main contributions of this dissertation and this chapter concludes in Section 1.5.

1.1 Previous Work

Modeling the relationship of multiple tasks is crucial in multi-task learning; all of the involved tasks are learnt simultaneously to improve the generalization performance of the resulting classifiers. In the literature, many approaches have been proposed to model the task relatedness from different perspectives, including sharing hidden units of neural networks among similar tasks [4,5], employing a common prior in hierarchical Bayesian models [6–9], learning multiple tasks with regularization and kernel methods [10,11], sharing parameters of Gaussian process [9,12,13], incorporating clustering for multi-task learning [14,15], and learning a shared feature mapping over the predictor space [16].

Sharing Hidden Nodes in Neural Network

Neural network has been well studied for learning multiple related tasks for improved generalization performance. Specifically, in [4, 5], the neural network based inductive bias learning models are considered for multi-task learning. Note that the inductive bias specifies a hypothesis for a learner (a learning algorithm) so that the hypothesis is large enough to contain an optimal solution to the learning problems of interest, yet small enough to guarantee reliable generalization performance over a pre-specified training set.

In [4], a neural network model is proposed for learning multiple related tasks, in which a set of hidden units are shared among multiple tasks for improved generalization, as illustrated in Figure 1.2. Specifically, in Figure 1.2 multiple tasks are fully connected to a shared hidden layer and they can select the useful hidden units by controlling the weights connecting to those hidden units. In essence the improved generalization performance (on each task) is achieved by using the training signals from other related tasks as the inductive bias, which facilitates the neural network model to select an optimal hypothesis for generalization.

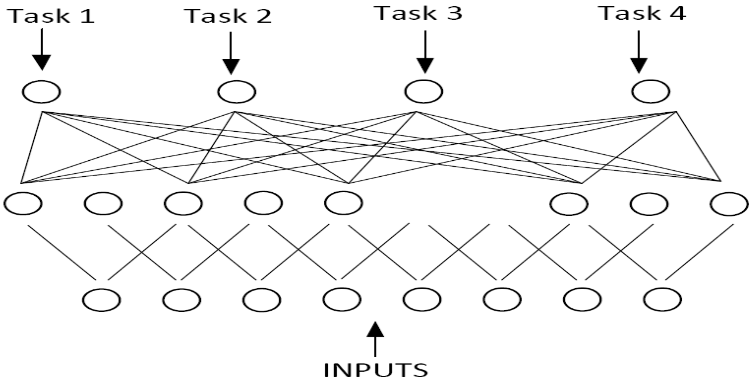


Figure 1.2: Multi-task backpropagation of four tasks: share a set of hidden nodes among multiple related tasks.

In [5], a similar neural network based bias learning model is proposed to automatically choose an optimal hypothesis space (from a family of hypothesis spaces) under the multi-task learning setting; moreover, a general concept of extended VC dimension is introduced and it is subsequently used to derive a generalization error bound to theoretically evaluate the performance of proposed bias learning model. The derived error bound demonstrates that learning multiple tasks simultaneously can potentially produce better generalization performance than learning a single task.

Constraining a Common Prior in Hierarchical Bayesian Models

Hierarchical Bayesian (HB) models have adaptive structures for modeling both the independence as well as the relatedness among multiple tasks. In HB models, the bottom layer of their hierarchies consist of a set of models with task-specific parameters, which are specialized for each task respectively; on the other hand, to capture the relatedness among the tasks, a commonly used approach is to correlate multiple tasks by constraining a common prior over the task-specific parameters of the bottom layer.

In tradition the common prior in a HB model is specified in parametric settings, that is, the HB model has a presumed parametric form while the model parameters are unknown. However, a parametric distribution may be too restrictive for modeling the common prior distribution, as the presumed parametric model may not reflect the reality of the learning scenarios. It is thus preferable to learn the functional form of the common prior from the training data directly, instead of specifying the functional form beforehand. In [6] and [7], different nonparametric HB based frameworks are proposed to unify the collaborative filtering (CF) and the content-based filtering; the proposed frameworks can automatically learn the common priors over the model parameters from multiple tasks (in the form of all user profiles). In [8], a probabilistic multi-task learning framework is proposed in which learning multiple tasks are treated as learning a Bayesian prior over the task space; specifically, the proposed framework identifies the latent independent components shared among the tasks, and then employ the identified components to capture the task relatedness. In [9], a novel HB based approach is proposed for learning the parameters of Gaussian processes under the multi-task learning setting; moreover insightful analysis is presented for the multi-task Gaussian process by exploiting the equivalence between the parametric linear model and the nonparametric Gaussian process.

Learning Multiple Tasks with Regularization and Kernel Methods

Regularization and Kernel methods are commonly used for modeling the task relatedness. In [10], a regularization framework is proposed for multi-task learning which enforces the individual task model close to the average of all task models. Specifically, the proposed framework considers the t-th task as a linear regression function as

$$f_t(x) = w_t^T x = (w_0 + v_t)^T x \approx y, \tag{1.1}$$

where v_t denotes the task-specific component (it is small when the tasks are similar) and w_0 denotes the average of all task models. By employing the hinge-loss function, the proposed regularized MTL formulation can be expressed as

$$\begin{aligned} \min_{w_0, v_t, \xi_{it}} \quad & \sum_{t=1}^{T} \sum_{i=1}^{m} \xi_{it} + \frac{\lambda_1}{T} \sum_{t=1}^{T} \|v_t\|^2 + \lambda_2 \|w_0\|^2 \\ \text{subject to} \quad & y_{it}(w_0 + v_t) \cdot x_{it} \geq 1 - \xi_{it},\ \xi_{it} \geq 0,\ i \in \mathbb{N}_m,\ t \in \mathbb{N}_T. \end{aligned} \tag{1.2}$$

For an arbitrary nonlinear feature map $\Phi : X \times \{1, \cdots, T\} \rightarrow \mathcal{H}$, where $\mathcal{H}$ is a separable Hilbert space (the feature space), the kernel associated to Φ is denoted as

$$G\left((x,t),(z,s)\right) = \langle \Phi(x,t), \Phi(z,s) \rangle . \tag{1.3}$$

Using the standard techniques, [10] extends the proposed MTL formulation to the non-linear setting as

$$\begin{aligned} \max_{\beta_i} \quad & \sum_{i=1}^{N} \beta_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i y_i \beta_j y_j G\left((x_i,t_i),(x_j,t_j)\right) \\ \text{subject to} \quad & 0 \leq \beta_j \leq C, i = 1, \cdots, N. \end{aligned} \tag{1.4}$$

Based on the MTL framework in [10], a family of multi-task kernel functions are developed in [11] to model the relations among multiple tasks (in the non-linear setting); moreover an interesting theoretical analysis is presented to show that learning multiple tasks with such family of kernels can be equivalently cast as single task learning problems.

Sharing Parameters of the Gaussian Process among Multiple Tasks

There is also an emerging interest in learning the kernel of the Gaussian Process under multi-task learning setting for both the regression and the classification. In [12] an efficient method is proposed to learn the parameters (of a shared covariance function) for the Gaussian process (GP). The proposed method adopts the multi-task informative vector machine (IVM) to greedily select the most informative examples from the separate tasks and hence alleviate the computation cost. Subsequently in [13], a novel approach is proposed to learn covariance matrices from multi-task data (input-dependent features and a free-form covariance matrix) via an EM-algorithm. One limitation in the proposed approach lies in that its generalization to new data could only be achieved by an ad-hoc form of kernel extrapolation.

Incorporating Clustering for Multi-Task Learning

Many MTL algorithms assume that the learning tasks are equally weighted/related (as depicted in the left plot of Figure 1.3). Learning the tasks simultaneously via modeling their relatedness generally improves the overall generalization. In reality, however, a group of tasks may be correlated, while some other tasks may be unrelated to such a group (as depicted in the right plot of Figure 1.3). Simply learning all tasks simultaneously (under the a single presumed MTL setting) may lead to sub-optimal performance.

In [14], a task clustering (TC) algorithm is proposed for discovering the clustering structure of multiple learning tasks. The TC algorithm estimates the mutual relatedness between tasks (via measuring the averaged generalization accuracy of the tasks using the knowledge borrowed from other tasks), and then builds up an entire hierarchy of previous tasks. When a new learning task arrives, the TC algorithm identifies the most related task cluster in the hierarchy and then apply the knowledge (a type of distance metric) from that cluster for learning the new task.

In [15], a convex formulation is proposed for clustered multi-task learning. The proposed formulation encodes the (unknown) task cluster information into a novel penalty norm. Formally, the

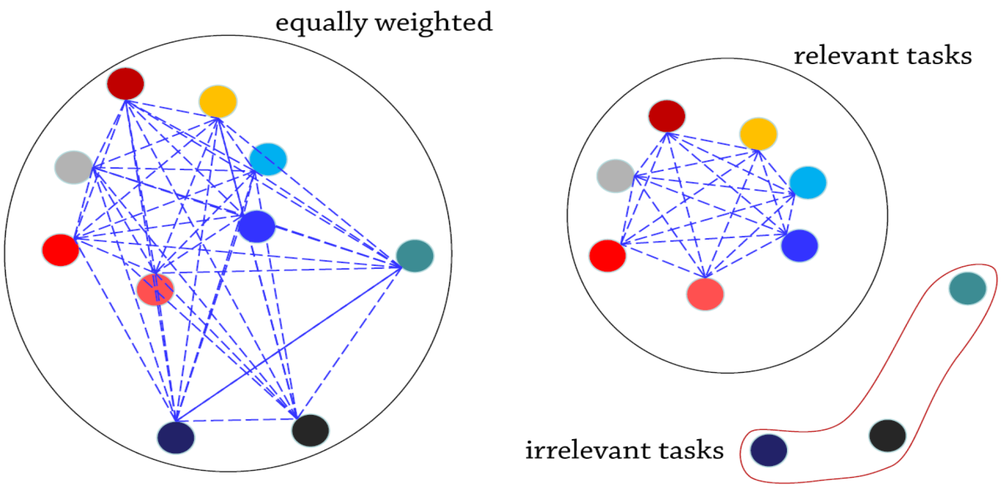


Figure 1.3: The illustration of the relationship among multiple tasks: all tasks are equally weighted (left plot); a group of tasks are correlated while some other tasks are irrelevant to such a group (right plot).

penalty norm is denoted as

$$\widehat{\Omega}(W) = \varepsilon_m \Omega_{\text{mean}} + \varepsilon_b \Omega_{\text{between}} + \varepsilon_w \Omega_{\text{within}}, \tag{1.5}$$

where Ω_{mean} measures how large the weight vectors are, Ω_{between} quantifies how close to each other the different clusters are, Ω_{within} quantifies the compactness of the clusters, ε_m, ε_b and ε_w correspond to the respective coefficients of the aforementioned terms. Therefore the proposed MTL formulation can be expressed as

$$\min_W \; \mathcal{L}_{\{X_t, Y_t\}}(W) + \widehat{\Omega}(W), \tag{1.6}$$

where $\mathcal{L}_{\{X_t, Y_t\}}(W)$ denotes the empirical loss over the training data $\{X_t, Y_t\}$. The proposed formulation in Eq. (1.6) is not convex; subsequently in [15], it is converted into a convex relaxation and an efficient algorithm is developed to find the globally optimum of the convex relaxation.

Learning a Shared Feature Mapping over the Predictor Space

Recently, there is a growing interest in learning a shared feature mapping from multiple related tasks [16]. Such a feature mapping generally corresponds to a low-rank structure shared among multiple tasks (represented by the weight vectors of the tasks).

In [16], an alternating structure optimization (ASO) formulation is proposed for learning a shared predictive structure from multiple related tasks, as depicted in Figure 1.4. In ASO, each of the task corresponds to a linear predictive functions $f_\ell(x) = u_\ell^T x$, where u_ℓ is the weight vector

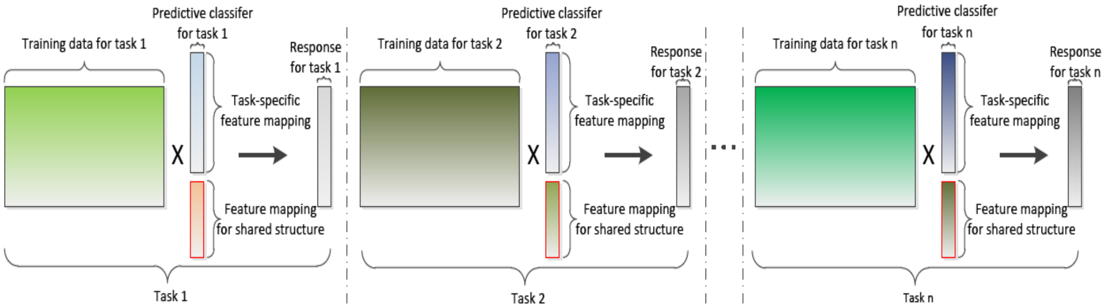

Predictive classifer for task 1
Training data for task 1
Response for task 1
X
Task-specific feature mapping
Feature mapping for shared structure
Task 1
Predictive classifer for task 2
Training data for task 2
Response for task 2
X
Task-specific feature mapping
Feature mapping for shared structure
Task 2
...
Predictive classifer for task n
Training data for task n
Response for task n
X
Task-specific feature mapping
Feature mapping for shared structure
Task n

Figure 1.4: Illustration of multi-task learning using a shared feature representation: the predictive classifier of each task consists of a task-specific feature mapping and a feature mapping of the shared structure.

for the ℓ-th task consisting of a task-specific component and a shared-among-tasks component. Specifically, u_ℓ can be expressed as

$$u_\ell = w_\ell + \Theta^T v_\ell, \;\; \Theta\Theta^T = I, \;\; \Theta \in \mathbb{R}^{h \times d}, \;\; h < d, \tag{1.7}$$

where u_ℓ, w_ℓ, and v_ℓ respectively represent the weight vectors for the full feature space, the high-dimensional feature space, and the shared low-dimensional feature space, and Θ represents the structure parameter for extracting the low-dimensional feature mapping. Note that ASO requires $d > h$, where h specifies the dimensionality of the extracted low-rank structure. Mathematically, ASO can be formulated as

$$\begin{aligned} \min_{\{u_\ell, v_\ell\}, \Theta} \quad & \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) + \alpha \|u_\ell - \Theta^T v_\ell\|^2 \right) \\ \text{subject to} \quad & \Theta\Theta^T = I_{h \times h}. \end{aligned} \tag{1.8}$$

The ASO formulation in Eq. (3.5) is non-convex; in [16] an alternating optimization algorithm is proposed to compute a local optimum of ASO.

1.2 Relation to other Learning Methods

Multi-task learning is a special case of *transfer Learning* [17] and it also subsumes *multi-label learning* [18] and *multi-class learning* [2] as special cases. We summarize the main difference of these methods below.

Transfer Learning

Transfer Learning subsumes multi-task learning as a special case. Transfer learning explicitly defines the source domain and the target domain; it aims at achieving better generalization performance in the target domain by transferring the domain knowledge learned from the source domain to target domain. Note that in transfer learning the feature space and the distribution of the samples from the source domain may be different from those from the target domain. When the source domain and the target domain coincide, transfer learning is equivalent to multi-task learning.

Multi-Task Learning

Multi-task learning subsumes multi-label learning as a special case. It aims at improving the overall generalization performance by learning multiple tasks simultaneously. The key component in multi-task is the modeling of the task relatedness. In multiple task learning, each task generally has different training samples. When all tasks share the same set of training data and features, multi-task learning is equivalent to multi-label learning.

Multi-Label Learning

Multi-label learning deals with the learning scenario where each sample is associated with multiple labels. Specifically, in multi-label learning, all labels share the same set of training data and features. When each sample is associated with a single label, multi-label learning is equivalent to multi-class learning.

1.3 Applications

Multi-task learning has been applied successfully in many application domains such as bioinformatics [19], image analysis [3, 20], web search ranking [21], and computer vision [22–24]. In this dissertation, we focus on two real-world applications, i.e., automated annotation of the *Drosophila* gene expression pattern images and categorization of the Yahoo web pages.

Automated Annotation of the Drosophila Gene Expression Pattern Images

The *Drosophila* gene expression pattern images capture the spatial and temporal dynamics of gene expression and hence facilitate the explication of the gene functions, interactions, and networks during *Drosophila* embryogenesis [25, 26]. To provide text-based pattern searching, the gene expression pattern images are annotated (according to their stage ranges) manually using a structured controlled vocabulary (CV) in small groups based on the genes and the developmental stages as shown in Figure 1.5. However, with a rapidly increasing number of gene expression pattern images, it is desirable to design computational approaches to automate the CV annotation process.

We preprocess the *Drosophila* gene expression pattern images (of the standard size 128×320) from the FlyExpress database following the procedures in [27]. The *Drosophila* images are from 16 specific stages, which are then grouped into 6 stage ranges ($1 \sim 3$, $4 \sim 6$, $7 \sim 8$, $9 \sim 10$, $11 \sim 12$, $13 \sim 16$). We manually annotate the image groups (based on the genes and the developmental stages) using the structured CV terms. Each image group is then represented as a feature vector based on the bag-of-words and the soft-assignment sparse coding schemes. Note that the SIFT (scale-invariant feature transform) features [28] are extracted from the images with the patch size set at 16×16 and the number of visual words in sparse coding set at 2000. The first stage range only contains 2 CV terms and we do not include it for our empirical study. For other stage ranges, we consider the top 10 and 20 CV terms that appear most frequently in the image groups and treat the annotation of each CV term as one task.

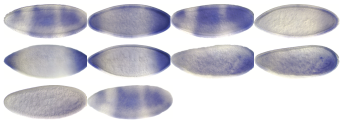
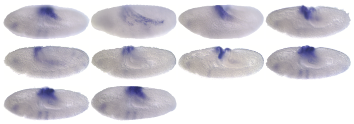
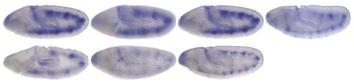
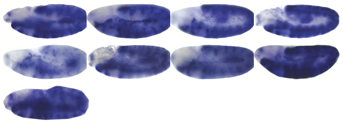
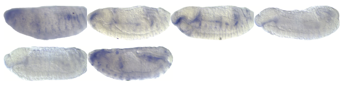
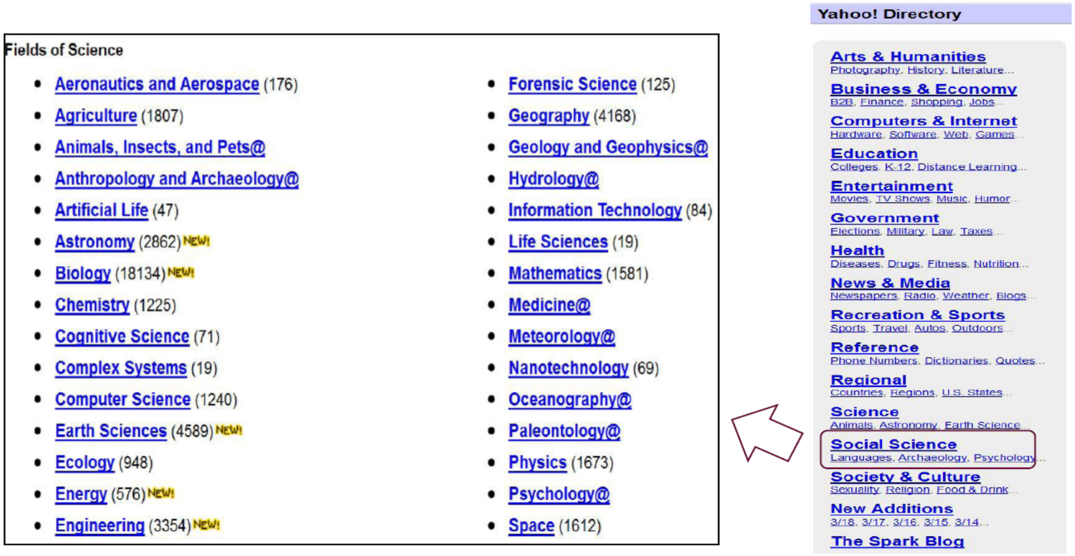
Stage range	Gene	Images Group	CV terms
$4 \sim 6$	Mkp3		cellular blastoderm clypeolabrum anlage in statu nascendi dorsal ectoderm anlage in statu nascendi endoderm anlage in statu nascendi foregut anlage in statu nascendi gap subset ventral ectoderm anlage in statu nascendi
$7 \sim 8$	dap		amnioserosa anlage ventral ectoderm primordium P2
$9 \sim 10$	W		inclusive hindgut primordium mesectoderm primordium procephalic ectoderm primordium trunk mesoderm primordium ventral ectoderm primordium
$11 \sim 12$	Ama		atrium primordium brain primordium clypeo-labral primordium dorsal epidermis primordium gnathal primordium head epidermis primordium P1 hindgut proper primordium midline primordium ventral epidermis primordium ventral nerve cord primordium
$13 \sim 16$	CG32048		atrium embryonic brain embryonic central nervous system embryonic dorsal epidermis embryonic epipharynx embryonic head epidermis embryonic large intestine embryonic ventral epidermis ventral midline ventral nerve cord

Figure 1.5: Sample image groups (from 5 different stage ranges) and their associated controlled vocabulary (CV) terms.

Categorization of the Yahoo Web Pages

The Yahoo directory [29] consists of a set of top-level categories such as *Arts & Humanities* and *Business & Economy*, as illustrated in the right plot of Figure 1.6. Each top-level category is further divided into a set of second-level sub-categories, where each second-level sub-category corresponds to a topic (one top-level category). For example, *Social Science* top-level category (the right plot of Figure 1.6) consists of 30 second-level categories (topics).

We apply the multi-task learning algorithms for the categorization of Yahoo webpages. Specifically we apply the multi-task learning algorithms on the webpages from the same top-category; since the webpages from different second-level categories (belonging to the same top-level category) share some commonality, the determination of the webpages for one second-level category is modeled as one task and hence the determination of the webpages fro multiple second-level categories are modeled as multiple related tasks. Note that we preprocess the Yahoo web pages by removing the topics with a small number (less than 100) of web pages; we also extract the TF-IDF (Term Frequency-Inverse Document Frequency) features from the web pages and the obtained feature vectors are normalized to unit length.



Fields of Science

- Aeronautics and Aerospace (176)
- Agriculture (1807)
- Animals, Insects, and Pets@
- Anthropology and Archaeology@
- Artificial Life (47)
- Astronomy (2862) NEW!
- Biology (18134) NEW!
- Chemistry (1225)
- Cognitive Science (71)
- Complex Systems (19)
- Computer Science (1240)
- Earth Sciences (4589) NEW!
- Ecology (948)
- Energy (576) NEW!
- Engineering (3354) NEW!
- Forensic Science (125)
- Geography (4168)
- Geology and Geophysics@
- Hydrology@
- Information Technology (84)
- Life Sciences (19)
- Mathematics (1581)
- Medicine@
- Meteorology@
- Nanotechnology (69)
- Oceanography@
- Paleontology@
- Physics (1673)
- Psychology@
- Space (1612)

Yahoo! Directory

Arts & Humanities
Photography, History, Literature...
Business & Economy
B2B, Finance, Shopping, Jobs...
Computers & Internet
Hardware, Software, Web, Games...
Education
Colleges, K-12, Distance Learning...
Entertainment
Movies, TV Shows, Music, Humor...
Government
Elections, Military, Law, Taxes...
Health
Diseases, Drugs, Fitness, Nutrition...
News & Media
Newspapers, Radio, Weather, Blogs...
Recreation & Sports
Sports, Travel, Autos, Outdoors...
Reference
Phone Numbers, Dictionaries, Quotes...
Regional
Countries, Regions, U.S. States...
Science
Animals, Astronomy, Earth Science...
Social Science
Languages, Archaeology, Psychology...
Society & Culture
Sexuality, Religion, Food & Drink...
New Additions
3/18, 3/17, 3/16, 3/15, 3/14...
The Spark Blog

Figure 1.6: Illustration of the Yahoo Webpages: the right plot represents 16 top-level categories; the left plot represents the 30 second-level categories from the *Social Science* top-level category.

1.4 Main Contributions

In this dissertation, I consider the learning scenario where multiple tasks are intrinsically related via a shared low-dimensional feature mapping.

I develop a series of MTL approaches which follow the strand of inducing a low-dimensional feature space via certain low-rank constraints. Specifically, the proposed MTL approaches are formulated as mathematical programming problems in a generic form of minimizing the (nonnegative) linear combination of empirical loss (over the training data) with different structured regularizations (arising from different problem settings). The employed structured regularizations include: (1) the trace norm constraint (for selecting a set of shared basis factors); (2) the combination of the orthonormal constraint (for inducing a shared low-rank feature mapping) and the ℓ_2-norm (for selecting an dependent feature mapping for each task); (3) the combination of the trace norm constraint (for inducing a shared low-rank feature mapping) and the sparse regularization (for selecting discriminative feature for each task) (4) the combination of the trace norm regularization (for inducing a shared low-rank feature mapping) and the group sparse regularization (for identifying the irrelevant tasks); (5) the sparse trace norm regularization (for inducing a simultaneously spare and low-rank mapping).

The proposed MTL formulations can be solved via many existing solvers, which may not scale to large scale data sets. I propose to apply two types of algorithms, i.e., the alternating optimization based algorithms and the gradient based algorithms, to find their globally optimal solution efficiently. Moreover, I develop efficient algorithms for solving the key components involved in the optimization algorithms. I also conduct theoretical analysis for certain MTL approaches such as deriving the global solution recovery condition and the performance bound.

To demonstrate the practical performance, I also apply the proposed MTL approaches on different real-world applications: (1) Automated annotation of the Drosophila gene expression pattern images; (2) Categorization of the Yahoo web pages. Our experimental results demonstrate the efficiency and effectiveness of the proposed algorithms.

1.5 Summary of the Remaining Chapters

[Chapter 2 - Factor Selection and Coefficient Estimation in Multivariate Linear Regression] In this chapter, I consider the factor estimation and selection (FES) for multiple related regression functions. I first formulate FES as a multivariate linear regression problem subject to a trace norm constraint.

I then propose to employe the gradient based scheme for solving the FES formulation and also develop an efficient algorithm for the key component of the employed gradient scheme. I finally present experimental results to demonstrate the efficiency and effectiveness of the proposed algorithms.

[Chapter 3 - Learning a Shared Low-Rank Structure from Multiple Tasks] In this chapter, I consider the problem of learning a shared low-dimensional feature mapping from multiple tasks. I first present an improved ASO formulation (iASO), which correlates multiple tasks via a low-rank structure. I then convert iASO, a non-convex formulation, into a relaxed convex one (rASO). The theoretical analysis shows that rASO finds a globally optimal solution to its non-convex counterpart iASO under certain conditions. I also propose efficient algorithms, namely gradient based algorithms and alternating based algorithms, to compute the optimal solution to rASO. Finally I report the experiments to demonstrate the effectiveness and efficiency of the proposed algorithms and confirm our theoretical analysis.

[Chapter 4 - Learning Incoherent Sparse and Low-rank Patterns from Multiple Tasks] In this chapter, I consider the problem of learning incoherent sparse and low-rank patterns from multiple tasks. I first propose a linear multi-task learning formulation, in which the sparse and low-rank patterns are induced by a sparse regularization term and a low-rank constraint, respectively. I then propose to employ the projected gradient scheme to efficiently solve the proposed formulation; I also develop efficient algorithms for solving the key components of the projected gradient based algorithms. In addition I discuss the rates of convergence of the proposed projected gradient based algorithms in details. Experimental results on a collection of real-world data sets demonstrate the effectiveness of the proposed multi-task learning formulation and the efficiency of the proposed projected gradient algorithms.

[Chapter 5 - Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning] In this chapter, I consider the scenarios where a group of tasks are related while the other tasks are irrelevant to such a group. I first propose a robust multi-task learning (RMTL) algorithm which learns multiple tasks simultaneously as well as identifies the irrelevant tasks. I then develop efficient optimization algorithms to solve the proposed RMTL formulation. I also theoretically analyze the effectiveness of the RMTL algorithm, i.e., derive a theoretical bound for characterizing the learning performance of RMTL. Our experimental results on benchmark data sets demonstrate the effectiveness and efficiency of the proposed algorithm.

[Chapter 6 - Learning Multiple Tasks via Sparse Trace Norm Regularization] In this chapter, I consider the problem of estimating multiple predictive functions from a dictionary of basis functions in the nonparametric regression setting. I first formulate the function estimation problem as a convex program regularized by the trace norm and the ℓ_1-norm simultaneously. I then develop efficient optimization algorithms to solve the convex program. In addition, I theoretically establish a performance bound for the proposed function estimation scheme. The simulation studies demonstrate the effectiveness and efficiency of the proposed algorithms.

[Chapter 7 - Conclusion and Future Directions] In this chapter, I provide a summary of the dissertation and discuss several future research directions.

Notations Denote $\mathbb{N}_n = \{1, \cdots, n\}$. Denote by $\mathbb{S}_+^d$ the subset of positive semidefinite matrices. For any symmetric matrix M, denote its trace by tr(M), and its inverse by M^{-1}. For any pair of matrices A and B, denote $A \preceq B$ if and only if $B - A$ is positive semidefinite. For any matrix $A = [a_1, \cdots, a_m] \in \mathbb{R}^{d \times m}$, let $a_i \in \mathbb{R}^d$ be the i-th column in A; let a_{ij} be the entry in the i-th row and j-th column in A; denote by $\|A\|_0$ the number of nonzero entries in A; let $\|A\|_1 = \sum_{i=1}^d \sum_{j=1}^m |a_{ij}|$; denote by $\|A\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^m a_{ij}^2}$ the Frobenius norm; let $\{\sigma_i(A)\}_{i=1}^r$ be the set of singular values of A in non-increasing order, where $r =$ rank(A); denote by $\|A\|_2 = \sigma_1(A)$ and $\|A\|_* = \sum_{i=1}^r \sigma_i(A)$ the operator norm and trace norm of A, respectively; let $\|A\|_\infty = \max_{i,j} |a_{ij}|$; denote by $\|a_i\|_2$ the ℓ_2-norm of a_i; let $\|A\|_{\infty,2} = \|a_j\|_2$, where $j = \arg\max_i \|a_i\|_2$; let $\|A\|_{1,2} = \sum_{i=1}^m \|a_i\|_2$; Denote by $I_{h \times h}$ the identity matrix of size h by h. For any smooth function $f(\cdot)$, denote its gradient at the point C by $\nabla f(C)$.

Chapter 2

Factor Selection and Coefficient Estimation in Multivariate Linear Regression

2.1 Introduction

Multivariate linear regression (MLR) has been used widely for modeling the predictive relationship of the observations and multiple related responses in many applications of machine learning and data mining [30–32]. The coefficient matrix in MLR can be computed using the classical least squares estimator, where each response is regressed against the observations separately. It is known that such an estimator performs sub-optimally without utilizing the correlation among the multiple responses [31]. Linear factor regression is proposed to overcome such a problem; the main idea is to regress the responses against the factors, i.e., a small set of transformed observations. However, in linear factor regression, the factor selection is performed in a separate step via hypothesis test or cross-validation, independent from the subsequent coefficient matrix estimation.

Recently, Yuan [33] proposed a novel linear factor regression model, named factor estimation and selection method (FES), to select the factors and estimate the coefficient matrix simultaneously. The FES algorithm is formulated as an MLR problem subject to a trace norm constraint [34, 35], which enforces a low-rank constraint on the coefficient matrix in MLR. The low-rank property is important in that it promotes sparsity in the factor space, thus FES estimates the coefficient matrix based on the shrinking factor space, and it performs factors selection and coefficient matrix estimation simultaneously.

Solving the FES formulation is, however, challenging in practice, due to the non-smoothness of the trace norm constraint. Fazel [36] and Srebro [37] formulated the optimization problems involving trace norm components as semidefinite programs (SDP) [38]. In [33, 39], the FES formulation was reformulated as second-order cone programs (SOCP). Both SDP and SOCP can be solved via interior point methods [40], in which the second order information, i.e., Hessian matrix, is required for the computation. Note that many off-the-shelf optimization solvers such as SeDuMi [41] and SDPT3 [42] can be used for solving SDP and SOCP, which can only handle several hundreds of optimization variables. However, in many real applications such as image deblurring, the optimization problems could be of large scale and involve dense data matrices, thus the use of the sophisticated interior points methods is often precluded.

First-order methods, such as (sub)gradient methods [38, 43] and Nesterove's first-order optimal method [44, 45], are practical options for large-scale optimization problems; they only require to evaluate the function value and the (sub)gradient at each iteration. Cai [46] proposed a first-order algorithm to approximate a matrix with the minimum trace norm subject to a set of convex constraints. Lu [39] further reformulated the FES formulation as a penalized least squares formulation and then applied Nesterov's first-order optimal method. However, such a reformulation leads to a non-smooth (non-differentiable) objective function and hence a special smooth approximation scheme is required [47].

We propose to apply the gradient scheme (first-order) for solving the FES formulation in the form of a constrained MLR problem. This consideration clearly leads to an easy-to-solve smooth objective function during the optimization process. The general step of the gradient scheme involves a nontrivial Euclidean projection procedure, in which the feasible solution point is updated from an auxiliary one constructed from previous iterations. We show that such a projection procedure can be formulated as a simple singular optimization problem, which can be solved efficiently via many existing algorithms [48]. We present a simple gradient method with the proposed efficient Euclidean projection for solving the FES formulation. It can be shown that such a simple method converges slowly at the rate of $\mathcal{O}\left(\frac{1}{k}\right)$. We accelerate the gradient method based on an algorithm developed by Nesterov [44, 45] for minimizing a smooth convex function. It can be shown that the accelerated gradient method converges at the optimal rate of $\mathcal{O}(\frac{1}{k^2})$ among first-order methods, and meanwhile keeps the simplicity of the gradient method. We conduct simulation on synthetic and real-world data sets. Experimental results demonstrate the efficiency and effectiveness of the proposed algorithms.

2.2 Factor Estimation and Selection

We are given n observations (samples) with p explanatory variables (feature dimensions) $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n \times p}$ and q responses (labels) $Y = [y_1, \cdots, y_n]^T \in \mathbb{R}^{n \times q}$, which are assumed to be related. MLR models the predictive relationship between the observations and multiple responses via the classical least squares estimator:

$$\min_{W \in \mathbb{R}^{p \times q}} \|XW - Y\|_F^2, \tag{2.1}$$

where W denotes the coefficient matrix. The computation of W in Eq. (2.1) is equivalent to regressing each response against the observations separately, which may perform sub-optimally without utilizing the correlation among multiple responses.

Linear factor regression models are used to overcome this problem, in which the responses are regressed against a selected set of factors, i.e., a small number of linearly transformed observations. Mathematically, the linear factor regression models can be expressed as:

$$\min_{r\in\mathbb{R},\Gamma\in\mathbb{R}^{p\times r},\Omega\in\mathbb{R}^{r\times q}} \|X\Gamma\Omega - Y\|_F^2, \tag{2.2}$$

where $r \leq \min(p,q)$ specifies the number of factors, and $X\Gamma \in \mathbb{R}^{n\times r}$ denotes the factors, and $\Omega \in \mathbb{R}^{r\times q}$ denotes the factor loadings. In practice, the value of r is determined first in a separate step through either hypothesis testing or cross-validation. Then Γ is constructed and Ω can be estimated via the least squares estimator. Many popular methods could be formulated in the form of linear factor regressions. These methods differ in how the factors are constructed.

Recently, Yuan [33] proposed the factor estimation and selection (FES) method to determine the number of factors r, the factor matrix Γ, and the factor loading matrix Ω simultaneously. Let $W = U_w\Sigma_w V_w^T$ be the full SVD [49] of W, where $U_w \in \mathbb{R}^{p\times p}$ and $V_w \in \mathbb{R}^{q\times q}$ are orthogonal, and $\Sigma_w \in \mathbb{R}^{p\times q}$ is diagonal consisting of singular values. By choosing $\Gamma = U_w$ and $\Omega = \Sigma_w V_w^T$ in Eq. (2.2), and bounding the sum of singular values of Ω from above using a pre-specified nonnegative constant m, FES can be expressed as a constrained least squares problem as:

$$\begin{aligned} \min_{W\in\mathbb{R}^{p\times q}} \quad & \|XW - Y\|_F^2 \\ \text{subject to} \quad & \|W\|_* \leq m. \end{aligned} \tag{2.3}$$

Note that $\|\Omega\|_* = \|W\|_*$ since U_w is orthogonal. The trace norm constraint in Eq. (2.3) encourages sparsity in the singular values of W and hence results in automatic selection and estimation in the factor spaces. Therefore, the FES method conducts factors selection and coefficient matrix estimation simultaneously in MLR.

Equivalent Simplification

The FES formulation in Eq. (2.3) depends on the sample size, which may incur intensive computation when dealing with large-scale data sets. Eq. (2.3) can be equivalently simplified as a sample-size-independent optimization problem. Note that the simplification procedure is also employed in [39].

Let $X = U_x\Sigma_x V_x^T$ be the SVD of X, where $r = \text{rank}(X)$, $U_x \in \mathbb{R}^{n\times r}$ is columnwise orthonormal, $V_x \in \mathbb{R}^{p\times p}$ is orthogonal, and $\Sigma_x \in \mathbb{R}^{r\times p}$ is diagonal consisting of non-zero singular values. We have

$$\begin{aligned} \|XW - Y\|_F^2 &= \text{tr}\left(W^T V_x \Sigma_x^T \Sigma_x V_x^T W - 2Y^T U_x \Sigma_x V_x^T W + Y^T Y\right) \\ &= \|\Sigma_x V_x^T W - U_x^T Y\|_F^2 - \|U_x^T Y\|_F^2 + \|Y\|_F^2. \end{aligned}$$

Therefore, minimizing $\|XW - Y\|_F^2$ in Eq. (2.3) is equivalent to minimizing $\|\Sigma_x V_x^T W - U_x^T Y\|_F^2$. Denote $Z = V_x^T W \in \mathbb{R}^{p \times q}$, $\Lambda = \Sigma_x \in \mathbb{R}^{r \times p}$, and $H = U_x^T Y \in \mathbb{R}^{r \times q}$. Noticing $\|W\|_* = \|Z\|_*$, we can equivalently rewrite the problem in Eq. (2.3) as:

$$\begin{aligned} \min_{Z \in \mathbb{R}^{p \times q}} \quad & \|\Lambda Z - H\|_F^2 \\ \text{subject to} \quad & \|Z\|_* \le m. \end{aligned} \tag{2.4}$$

Since $\Lambda \in \mathbb{R}^{r \times p}$ is diagonal and the number of non-zero entries is smaller than $\min(n, p)$, from a computational point of view, the formulation in Eq. (2.4) needs less storage space and is easier to solve compared to the one in Eq. (2.3). Moreover, the optimal solution to Eq. (2.3) can be easily recovered from the one to Eq. (2.4) by applying an orthogonal transformation.

2.3 Gradient Scheme with Efficient Projection

In this section, we propose to apply the gradient scheme [44, 45] with efficient Euclidean projection to solve the constrained optimization problem in Eq. (2.4). For notational simplicity, we denote the optimization problem in Eq. (2.4) as

$$\begin{aligned} \min_{Z} \quad & f(Z) \\ \text{subject to} \quad & Z \in \mathcal{Q}, \end{aligned} \tag{2.5}$$

where $\mathcal{Q} = \{Z \mid \|Z\|_* \le m, Z \in \mathbb{R}^{p \times q}\}$ is a convex set, and $f(Z) = \|\Lambda Z - H\|_F^2$ is convex and continuously differentiable with Lipschitz continuous gradient $\mathcal{L}$ defined as [44, 45]:

$$\|\nabla f(Z_x) - \nabla f(Z_y)\|_F \le \mathcal{L}\|Z_x - Z_y\|_F, \forall Z_x, Z_y \in \mathcal{Q}. \tag{2.6}$$

To solve the constrained optimization problem in Eq. (2.5), the gradient scheme iteratively updates the feasible solution point Z via the general step denoted as:

$$Z = S - \frac{1}{\gamma} g_{S,\gamma}, \tag{2.7}$$

where $g_{S,\gamma}$ and $\frac{1}{\gamma}$ are the *gradient mapping* and the *step size* respectively, and S can be either a feasible solution point obtained in the last iteration or an auxiliary searching point (not necessarily feasible) constructed from pervious iterations.

The computation of $g_{S,\gamma}$ and γ in Eq. (4.9) are closely related to the *Euclidean projection* problem. In the following subsections, we first introduce some basic concepts of gradient mapping and *appropriate step size*, and then propose an efficient Euclidean projection algorithm associated with the optimization problem in Eq. (2.5).

Gradient Mapping

Gradient mapping [44,45] plays a central role in constrained (convex) optimization. Given the smooth convex function $f(Z)$ and the closed and bounded convex set $\mathcal{Q}$, we define the function $f_{S,\gamma}(Z)$ as:

$$f_{S,\gamma}(Z) = f(S) + \langle \nabla f(Z), Z - S\rangle + \frac{\gamma}{2}\|Z - S\|_F^2, \tag{2.8}$$

where $\gamma > 0$, $Z \in \mathcal{Q}$, and $\langle C_1, C_2\rangle = \text{tr}\left(C_1^T C_2\right)$. Note that S is not necessarily from $\mathcal{Q}$. Since $f_{S,\gamma}(Z)$ is smooth and strictly convex, its minimizer over $\mathcal{Q}$ is unique and can be denoted as:

$$Z_{S,\gamma} = \arg\min_{Z\in\mathcal{Q}} f_{S,\gamma}(Z). \tag{2.9}$$

The gradient mapping associated with $f(Z)$ on $\mathcal{Q}$ is then defined as:

$$g_{S,\gamma} = \gamma\left(S - Z_{S,\gamma}\right). \tag{2.10}$$

For a fixed S, we say that γ is appropriate (hence the global convergence can be guaranteed) if it satisfies

$$f(Z_{S,\gamma}) \leq f_{S,\gamma}(Z_{S,\gamma}),$$

where $f_{S,\gamma}(Z)$ is defined in Eq. (2.8) and $Z_{S,\gamma}$ is computed from Eq. (2.9). In practice, the appropriate γ can be determined using the sophisticated inexact line search algorithms, such as Armijo-Goldstein conditions [43]. Note that it can be shown [44] that any $\gamma \geq \mathcal{L}$ is appropriate, where $\mathcal{L}$ is the Lipschitz continuous gradient defined in Eq. (2.6). Moreover, for any appropriate γ, the following inequality holds [44, 45]

$$f(Z) \geq f(Z_{S,\gamma}) + \langle g_{S,\gamma}, Z - S\rangle + \frac{1}{2\gamma}\|g_{S,\gamma}\|_F^2. \tag{2.11}$$

for all $Z \in \mathcal{Q}$. Note that Eq. (2.11) is important for the global convergence analysis of constrained optimization problems.

Efficient Euclidean Projection

It can be verified that the optimal $Z_{S,\gamma}$ to Eq. (2.9) can be obtained by solving the following optimization problem:

$$\min_{Z\in\mathcal{Q}} \quad \left\|Z - \left(S - \tfrac{1}{\gamma}\nabla f(S)\right)\right\|_F^2. \tag{2.12}$$

Therefore, the computation of the gradient mapping and the step size can be cast as the Euclidean projection problem in Eq. (2.12). We propose an efficient algorithm to solve the following general

Euclidean projection problem:

$$\begin{aligned} \min_{Z\in\mathbb{R}^{p\times q}} \quad & \frac{1}{2}\|Z-\hat{S}\|_F^2 \\ \text{subject to} \quad & \|Z\|_* \le m, \end{aligned} \tag{2.13}$$

where m is a pre-specified non-negative constant, and $\frac{1}{2}$ is added to the objective function for easy calculation.

The efficient projection algorithm is devised based on the subgradients [44] of the Lagrangian function associated with the optimization problem in Eq. (2.13). Recall that for any non-smooth convex function $f : \mathbb{R}^{p\times q} \to \mathbb{R}$, the subgradient, $\partial f(\tilde{Z})$, of $f(Z)$ at the point $\tilde{Z} \in \mathbb{R}^{p\times q}$ is a compact convex set given by

$$\left\{G \in \mathbb{R}^{p\times q} : f(Z) \ge f(\tilde{Z}) + \text{tr}\left(G^T(Z-\tilde{Z})\right), \forall Z \in \mathbb{R}^{p\times q}\right\}.$$

Let $\tilde{Z} = P_{\tilde{z}}\Sigma_{\tilde{z}}Q_{\tilde{z}}^T$ be the thin SVD of $\tilde{Z}$, where $P_{\tilde{z}} \in \mathbb{R}^{p\times r}$ and $Q_{\tilde{z}} \in \mathbb{R}^{q\times r}$ are columnwise orthonormal, and $\Sigma_{\tilde{z}} \in \mathbb{R}^{r\times r}$ is diagonal consisting of the non-zero singular values. The subgradients of $\|Z\|_*$ at the point $\tilde{Z}$ is given by

$$\partial\|\tilde{Z}\|_* = \left\{P_{\tilde{z}}Q_{\tilde{z}}^T + D : D \in \mathbb{R}^{p\times q}, P_{\tilde{z}}^T D = 0,\, D_{\tilde{z}}Q_{\tilde{z}}^T = 0,\, \|D\|_2 \le 1\right\}. \tag{2.14}$$

The main result of this subsection is summarized in the following the theorem.

Theorem 2.3.1. *For any $\hat{S} \in \mathbb{R}^{p\times q}$, denote its full SVD by $\hat{S} = P_{\hat{s}}\Sigma_{\hat{s}}Q_{\hat{s}}^T$, where $P_{\hat{s}} \in \mathbb{R}^{p\times p}$ and $Q_{\hat{s}}^T \in \mathbb{R}^{q\times q}$ are orthogonal, and $\Sigma_{\hat{s}} \in \mathbb{R}^{p\times q}$ is diagonal consisting of the singular values. Then the optimal Z^* to Eq. (2.13) satisfies $Z^* = P_{\hat{s}}\Sigma_{z^*}Q_{\hat{s}}^T$, where $\Sigma_{z^*} \in \mathbb{R}^{p\times q}$ is diagonal with the singular values of Z^* on its main diagonal.*

Proof. Since the point **0** is strictly feasible for the problem in Eq. (2.13), Slater's condition is satisfied and strong duality holds [38]. Define the Lagrangian function $L(Z,\lambda)$ associated with Eq. (2.13) as:

$$L(Z,\lambda) = \frac{1}{2}\|Z-\hat{S}\|_F^2 + \lambda\left(\|Z\|_* - m\right),$$

where $\lambda \ge 0$ is the Lagrangian multiplier (the dual variable). Let Z^* and λ^* be optimal to Eq. (2.13). It follows that

$$Z^* = \arg\min_{Z} L(Z,\lambda^*), \tag{2.15}$$

where $L(Z,\lambda^*)$ is non-smooth due to the trace norm component. It is known that Z^* is optimal to Eq. (2.15) if and only if **0** is a subgradient of $L(Z,\lambda^*)$ at the point Z^*, that is,

$$\mathbf{0} \in \partial L(Z^*,\lambda^*) = Z^* - \hat{S} + \lambda^*\partial\|Z^*\|_*. \tag{2.16}$$

Let $Z^* = P_{z^*}\Sigma_{z^*}Q_{z^*}^T$ be the thin SVD of Z^*, where $P_{z^*} \in \mathbb{R}^{p\times r}$ and $Q_{z^*} \in \mathbb{R}^{q\times r}$ are columnwise orthonormal, $\Sigma_{z^*} \in \mathbb{R}^{r\times r}$ is diagonal with positive singular values. Let $P_{z^*}^{\perp} \in \mathbb{R}^{p\times(p-r)}$ and $Q_{z^*}^{\perp} \in \mathbb{R}^{q\times(q-r)}$ be the null space of P_{z^*} and Q_{z^*}, respectively. It follows from Eq. (2.14) that there exists a point $D_{z^*} = P_{z^*}^{\perp}\Sigma_d\left(Q_{z^*}^{\perp}\right)^T \in \mathbb{R}^{p\times q}$ such that

$$P_{z^*}Q_{z^*}^T + D_{z^*} \in \partial\|Z^*\|_*$$

satisfies Eq. (2.16). Note that $\Sigma_d \in \mathbb{R}^{(p-r)\times(q-r)}$ is diagonal consisting of the singular values of D_{z^*}. It follows that

$$\begin{aligned}\hat{S} &= Z^* + \lambda^*\left(P_{z^*}Q_{z^*}^T + D_{z^*}\right)\\ &= P_{z^*}\Sigma_{z^*}Q_{z^*}^T + \lambda^*\left(P_{z^*}Q_{z^*}^T + P_{z^*}^{\perp}\Sigma_d\left(Q_{z^*}^{\perp}\right)^T\right)\\ &= P_{z^*}(\Sigma_{z^*}+\lambda^* I)Q_{z^*}^T + P_{z^*}^{\perp}\left(\lambda^*\Sigma_W\right)\left(Q_{z^*}^{\perp}\right)^T\end{aligned}$$

corresponding to an SVD decomposition of $\hat{S}$. This completes the proof of this theorem. □

One immediate consequence of Theorem 2.3.1 is that the Euclidean projection problem in Eq. (2.13) can be reformulated as a simple singular value optimization problem. We conclude this section with the following lemma.

Lemma 2.3.1. *Let $\Sigma_{\hat{s}} \in \mathbb{R}^{p\times q}$ and $\Sigma_{z^*} \in \mathbb{R}^{p\times q}$ be defined as in Theorem 2.3.1, and assume rank$(\hat{S}) = r$ and $\Sigma_{\hat{s}} =$ diag$(\sigma_1,\cdots,\sigma_r,\mathbf{0})$. Then $\Sigma_{z^*} =$ diag$(\tau_1,\cdots,\tau_r,\mathbf{0})$, where $\{\tau_i\}_{i=1}^r$ can be computed by solving*

$$\begin{aligned}\min_{\tau_i} \quad & \sum_{i=1}^{r}(\tau_i-\sigma_i)^2\\ \textit{subject to} \quad & \sum_{i=1}^{r}\tau_i \le m,\ \ 0\le\tau_i.\end{aligned} \tag{2.17}$$

The problem in Eq. (2.17) can be efficiently solved via a similar algorithm as the one in [48] proposed for solving Euclidean projection onto the simplex.

2.4 Algorithms of the Gradient Methods

We propose to solve the optimization problem in Eq. (2.5) using two gradient methods with the efficient projection procedure from the last section, and analyze the rate of convergence.

Projected Gradient Method

We first propose a simple projected gradient method to solve the optimization problem in Eq. (2.5). Let Z_i be the feasible solution obtained in the i-th iteration. The projected gradient method minimizes

```
Input: $Z_0 \in \mathbb{R}^{p \times q}$, $\gamma_0 \in \mathbb{R}$, and max-iter.
Output: $Z \in \mathbb{R}^{p \times q}$.
for $i = 0, 1, \cdots$, max-iter do
    Compute $\nabla f(Z_i) = \Lambda^T \Lambda Z_i - \Lambda^T H$.
    while (true)
        Compute $A = Z_i - \nabla f(Z_i)/\gamma_i$.
        Compute thin SVD of $A$ as $A = U_a \Sigma_a V_a^T$.
        Compute $\Sigma_{\hat{z}}$ using Eq. (2.17) and $\hat{Z} = U_a \Sigma_{\hat{z}} V_a^T$.
        if $f(\hat{Z}) \leq f_{Z_i, \gamma_i}(\hat{Z})$ then exit the loop.
            else update $\gamma_i = \gamma_i \times 2$.
        end-if
    end-while
    Update $Z_{i+1} = \hat{Z}$ and $\gamma_{i+1} = \gamma_i$.
    if stopping criteria satisfied then exit the loop.
end-for
Set $Z = Z_{i+1}$.
```

Algorithm 1: FES via Projected Gradient Method

the objective function by iteratively updating the feasible solution via the general step as:

$$Z_{i+1} = \mathcal{P}_{\mathcal{Q}} \left(\mathcal{G} \left(Z_i \right) \right),$$

where $\mathcal{G}(Z_i) = Z_i - \frac{1}{\gamma} \nabla f(Z_i)$ denotes a gradient step on the feasible solution Z_i, and $\mathcal{P}_{\mathcal{Q}}$ denotes the Euclidean projection defined in Eq. (2.13); meanwhile it determines the appropriate step size $\frac{1}{\gamma}$ (via linear search) by ensuring

$$f(Z_{i+1}) \leq f_{Z_i, \gamma}(Z_{i+1}),$$

where $f_{Z_i, \gamma}$ is defined in Eq. (2.8). The pseudo-code of the projected gradient method is presented in Algorithm 1, and its convergence rate analysis is summarized in the following theorem (a similar proof can be found in [44]).

Theorem 2.4.1. *Let Z^* be the global minimizer to Eq. (2.5). Denote the number of iteration by k. Algorithm 1 converges at the rate of $\mathcal{O}(1/k)$, that is, for all $k \geq 1$, we have*

$$f(Z_k) - f(Z^*) \leq \frac{\hat{\gamma}}{2k} \| Z_0 - Z^* \|_F^2.$$

where $\hat{\gamma} = \max\{\gamma_0, 2\mathcal{L}\}$, and γ_0 and Z_0 are the pre-specified initial values for γ_i and Z_i in Algorithm 1 respectively, and $\mathcal{L}$ is the Lipschitz continuous gradient in Eq. (2.6).

Accelerated Gradient Method

The proposed projected gradient method from the last subsection is simple to implement, but converges slowly. We accelerate the gradient method by using an algorithm developed by Nesterov [45], which is shown to be an optimal first-order method for minimizing a smooth convex function [44].

```
Input: $Z_0 \in \mathbb{R}^{p\times q}$, $\gamma_0 \in \mathbb{R}$, and max-iter.
Output: $Z \in \mathbb{R}^{p\times q}$.
Set $Z_1 = Z_0$, $t_{-1} = 0$, $t_0 = 1$ and $\gamma_1 = \gamma_0$.
for $i = 1, 2, \cdots$, max-iter do
    Compute $\alpha_i = (t_{i-2} - 1)/t_{i-1}$.
    Compute $S_i = (1+\alpha_i)Z_i - \alpha_i Z_{i-1}$ and $\nabla f(S_i)$.
    while (true)
        Compute $A = S_i - \nabla f(S_i)/\gamma_i$.
        Compute thin SVD of $A$ as $A = U_a \Sigma_a V_a^T$.
        Compute $\Sigma_{\hat{z}}$ using Eq. (2.17) and $\hat{Z} = U_a \Sigma_{\hat{z}} V_a^T$.
        if $f(\hat{Z}) \leq f_{S_i,\gamma_i}(\hat{Z})$ then exit the loop
            else update $\gamma_i = \gamma_i \times 2$.
        end-if
    end-while
    Update $Z_{i+1} = \hat{Z}$ and $\gamma_{i+1} = \gamma_i$.
    if stopping criteria satisfied then exit the loop.
    Update $t_i = \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$.
end-for
Set $Z = Z_{i+1}$.
```

Algorithm 2: FES via Accelerated Gradient Method

Nesterov's method utilizes two sequences: (feasible) solution sequence $\{Z_i\}$ and searching point sequence $\{S_i\}$; in the i-th iteration, it computes the searching point as:

$$S_i = (1+\alpha_i)Z_i - \alpha_i Z_{i-1},$$

where the parameter $\alpha_i > 0$ can be appropriately determined from the algorithm; similar to the projected gradient method, it then updates the feasible solution via the general step as:

$$Z_{i+1} = \mathcal{P}_{\mathcal{Q}}\left(\mathcal{G}(S_i)\right),$$

and meanwhile determines the step size by ensuring

$$f(Z_{i+1}) \leq f_{S_i,\gamma}(Z_{i+1}).$$

Note that the searching point S_i may not be feasible for the optimization problem, which can be seen as a forecast of the next feasible solution point and hence leads to the faster convergence rate. The pseudo-code of the accelerated gradient method is presented in Algorithm 2, and its convergence rate analysis is summarized in the following theorem (the detailed proof can be found in [44]):

Theorem 2.4.2. *Let Z^* be the global minimizer to Eq. (2.5). Denote the number of iteration by k. Algorithm 2 converges at the rate of $\mathcal{O}(1/k^2)$, that is, for all $k \geq 1$, we have*

$$f(Z_k) - f(Z^*) \leq \frac{2\hat{\gamma}}{k^2}\|Z_0 - Z^*\|_F^2, \tag{2.18}$$

where $\hat{\gamma} = \max(\gamma_0, 2\mathcal{L})$, and γ_0 and Z_0 are the pre-specified initial values for γ_i and Z_i in Algorithm 2 respectively, and $\mathcal{L}$ is the Lipschitz continuous gradient in Eq. (2.6).

2.5 Experimental Results

In this section, we empirically investigate the performance of the proposed projected gradient method (PG) and the accelerated gradient method (AG) (with efficient Euclidean projection) on a collection of synthetic and real-world data sets. We also compare the FES algorithm with other representative ones in terms of classification accuracy.

Efficiency Comparison

We compare the methods PG and AG with SM in [39] on solving the FES formulation in terms of computation time (in seconds), iteration number, and optimized objective value. Note that SM applies Nesterov's optimal smooth method for solving the dual of the FES formulation (in the form of a max-min optimization problem) and obtains an approximate solution; moreover, it can only handle the cases where the sample size is larger than the feature dimension. We generate the synthetic data sets using the similar scheme as in [39]. SM terminates once its duality gap is less than 10^{-8}, and PG and AG terminate once they attain an objective value equal to or smaller than that of SM.

Table 2.1: Comparison of SM, AG, and PG in terms of iterations, computation time in seconds, and objective value of $\|\Lambda Z - H\|_F^2$ on synthetical data (sample size n, feature dimension p, and label number q). "-" stands for "not available" since SM cannot handle the the cases of $n < p$; for these cases, we terminate AG and PG once the change of the objective value is smaller than 10^{-8}.

Data Set	Iteration Number			Computation Time			Objective Value		
(n, p, q)	SM	AG	PG	SM	AG	PG	SM	AG	PG
(100, 20, 10)	33823	2700	3282	174.1	3.5	4.6	1.4743138673	1.4743138673	1.4743138673
(200, 40, 20)	41942	4405	5334	405.9	18.1	22.8	1.6604408368	1.6604408368	1.6604408368
(300, 60, 30)	39193	10116	13072	516.1	83.5	113.6	1.5695795380	1.5695795380	1.5695795380
(400, 80, 40)	36810	7712	9500	662.3	113.6	141.3	1.5957001358	1.5957001358	1.5957001358
(40, 500, 50)	-	367	1335	-	15.7	39.6	-	132.19246110	132.19245439
(100, 1000, 50)	-	338	2633	-	22.9	117.6	-	354.76303593	354.76283256
(150, 1500, 50)	-	532	3402	-	59.6	240.8	-	538.33683255	538.33674089
(200, 2000, 50)	-	480	5087	-	83.9	534.3	-	721.24953544	721.24909047

From the experimental results in Table 2.1, we can observe that AG and PG outperform SM substantially, where AG has the smallest iteration number and computation time among the three competing methods. Note that SM pre-computes the Lipschitz constant in the implementation, which affects the practical step size of the optimization scheme; in constrast PG and AG applies line search to compute an appropriate step size in each of the iterations.

Sensitivity Study

We conduct sensitivity study on the methods PG and AG using USPS data (sample size 3000, feature dimension 256, and label number 10). PG and AG terminate once the stopping criterion is satisfied, i.e., the change of the objective value in two successive iterations is smaller than 10^{-8}. From the plot (a) in Figure 3.1, we can observe that AG converges much faster than PG, and the convergence curves are consistent with the respective theoretical convergence rate order of AG ($\mathcal{O}(\frac{1}{k^2})$) and PG ($\mathcal{O}(\frac{1}{k})$). From the plot (b) in Figure 3.1, we can observe that AG requires less computation time than PG when using the same stopping criterion.

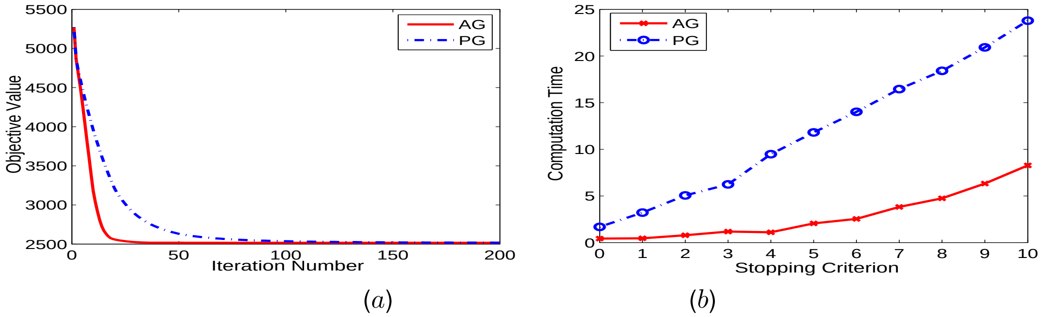


Figure 2.1: Sensitivity study on PG and AG: (a) objective value with respect to iterations; (b) computation time in seconds with respect to the stopping criterion (the difference of objective value in two successive steps); the index on x-axis denotes the exponent of the stopping criterion, i.e., 10^{-x}.

Performance Evaluation

We compare the FES algorithm with least squares (LS), principal component regression (PCR), and ridge regression (RR) in terms of classification accuracy on (sampled) real-world data sets[1], including Satimage (sample size 6435, feature dimension 36, label number 6), PIE ($1500 \times 124 \times 10$), USPS ($3000 \times 256 \times 10$), Soybean ($562 \times 35 \times 15$) and Letter ($20000 \times 26 \times 26$). We set the training ratio at 40%, and employed 1-NN as the classifier. The parameters in FES, PCR and RR are determined via cross-validation. The classification accuracy averaged over 10 random repetitions as well as the standard deviation is presented in Table 2.2.

Table 2.2: Classification accuracy (in percentage) comparison of four competing algorithms on real-word data sets.

Dataset	LS	PCR	RR	FES
Satimage	84.28 ± 0.37	83.04 ± 0.53	84.43 ± 0.37	85.77 ± 0.31
PIE	96.02 ± 0.60	96.87 ± 1.06	96.04 ± 0.56	98.18 ± 0.53
USPS	87.58 ± 0.48	89.48 ± 0.75	90.99 ± 0.77	91.14 ± 0.54
Soybean	90.92 ± 1.16	90.74 ± 1.97	90.98 ± 1.56	91.87 ± 1.02
Letter	93.08 ± 0.84	92.84 ± 1.18	94.29 ± 0.84	94.27 ± 0.86

From Table 2.2, we can observe that FES outperforms other competing algorithms on Satimage, PIE, USPS and Soybean; FES and RR perform competitively on Letter. This demon-

[1] http://archive.ics.uci.edu/ml/

strates the effectiveness of FES in capturing the correlation among multiple responses via the trace norm constraint and hence improving the classification accuracy.

2.6 Summary

We propose gradient projection methods for solving the factor selection and estimation (FES) formulation in the form of a multivariate linear regression problem subject to a trace norm constraint. We show that the nontrivial Euclidean projection (based on trace norm) in the gradient scheme can be reformulated as a simple singular value optimization problem, and hence can be solved efficiently. We present a simple gradient method for solving the FES formulation, and then accelerate it using Nesterov's first-order optimal algorithm. We empirically demonstrate the efficiency of the proposed gradient methods in comparison with the SM method in [39], and conduct sensitivity study on them. The FES formulation can be suitably extended to the case where each response corresponds to a unique set of observations. We plan to apply FES to multi-task problems, in which different tasks are provided with different sets of training data.

Chapter 3

Learning A Shared Low-Rank Structure from Multiple Tasks

3.1 Introduction

Recently, there has been a growing interest in studying multi-task learning in the context of feature learning (selection). Jebara [50] considered the problem of feature selection with SVM across the tasks. Obozinski [51] presented a work of multi-task joint covariate selection based on a generalization of 1-norm regularization. Argyriou [52] proposed to learn a common sparse representation from multiple tasks, which can be solved via an alternating optimization algorithm. One following work in [53] proposed the convex multi-task feature learning formulation and showed that the alternating optimization algorithm converges to a global optimum of the proposed formulation. Note that the MTL formulation in [53] is essentially equivalent to the approach of employing the trace norm as a regularization for multi-task learning [54–56]. Ando and Zhang [16] proposed the alternating structure optimization (ASO) to learn shared predictive structures from multiple related tasks. In ASO, a separate linear classifier is trained for each task and dimension reduction is applied on the classifier space, computing low-dimensional structures with the highest predictive power. However, this framework is non-convex and the alternating structure optimization procedure is not guaranteed to find a global optimum as pointed out in [16, 53].

We consider the problem of learning a shared structure from multiple related tasks following the approach in [16]. We present an improved ASO formulation (called iASO) using a new regularizer. The improved formulation is non-convex; we show that it can be converted into a relaxed convex formulation (called rASO). In addition, we present a theoretical condition, under which rASO finds a globally optimal solution to its nonconvex counterpart iASO. rASO can be equivalently reformulated as a semidefinite program (SDP) and solved via many off-the-shelf optimization solvers. However, SDP is not scalable to large data sets due to its positive semidefinite constraints.

We propose to employ the accelerated projected gradient (APG) algorithm to solve rASO. APG belongs to the category of the first-order methods and its global convergence rate is optimal among all first-order methods [45, 57]. We show that the subproblem in each iteration of APG can be solved efficiently. We further show that the computational cost of APG mainly depends on the feature dimensionality. We also develop the convex alternating structure optimization (CASO) algorithm to solve rASO. CASO is similar in spirit to the block coordinate descent method [58]. In CASO, the optimization variables are optimized via two alternating computation procedures; we develop efficient algorithms for the procedures in CASO and show that the algorithm converges to a global optimum of rASO. We show that the computational cost in CASO mainly depends on the sample size of the training data. We have conducted experiments on the yahoo web pages data sets [29] and the *Drosophila* gene expression pattern images[1] data sets. The experimental results demonstrate the effectiveness of the proposed MTL formulation and the efficiency of the proposed optimization algorithms. Results also confirm our theoretical analysis, i.e., rASO finds a globally optimal solution to its non-convex counterpart iASO under certain conditions.

The chapter is organized as follows: in Section 3.2 we present the improved MTL formulation iASO; in Section 3.3 we show how to convert the non-convex iASO into the convex relaxation rASO; in Sections 3.4 and 3.5, we detail the APG algorithm and the CASO algorithm respectively for solving rASO; in Section 3.6 we present a theoretical condition under which a globally optimal solution to iASO can be obtained via rASO; we report the experimental results in Section 3.8 and this chapter concludes in Section 4.8.

3.2 Multi-Task Learning Framework

Assume that we are given m supervised (binary-class) learning tasks. Each of the learning tasks is associated with a set of training data

$$\{(x_1^\ell, y_1^\ell), \cdots, (x_{n_\ell}^\ell, y_{n_\ell}^\ell)\} \subset \mathbb{R}^d \times \{-1, 1\},\ \ell \in \mathbb{N}_m, \tag{3.1}$$

and a linear predictor f_ℓ

$$f_\ell(x) = u_\ell^T x,\ \ell \in \mathbb{N}_m, \tag{3.2}$$

where u_ℓ is the weight vector for the ℓth task.

The alternating structure optimization (ASO) algorithm learns predictive functional structures from multiple related tasks. Specifically, it learns all m predictors $\{f_1, f_2, \cdots, f_m\}$ simultaneously by exploiting a shared feature space in a simple linear form of low-dimensional feature map Θ across the m tasks. Formally, the predictor f_ℓ can be expressed as:

$$f_\ell(x) = u_\ell^T x = w_\ell^T x + v_\ell^T \Theta x, \tag{3.3}$$

where the structure parameter Θ takes the form of an $h \times d$ matrix with orthonormal rows as

$$\Theta\Theta^T = I_{h\times h}, \tag{3.4}$$

[1] http://www.flyexpress.net/

and u_ℓ, w_ℓ, and v_ℓ are the weight vectors for the full feature space, the high-dimensional feature space, and the shared low-dimensional feature space, respectively. Note that since h specifies the shared low-dimensional feature space of the m tasks, in general we have $h \leq \min(m, d)$. Mathematically, ASO can be formulated as the following optimization problem:

$$\begin{array}{ll} \min_{\{u_\ell, v_\ell\}, \Theta} & \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) + \alpha \|w_\ell\|^2 \right) \\ \text{subject to} & \Theta\Theta^T = I_{h \times h}, \end{array} \quad (3.5)$$

where L is a convex loss function, $\|w_\ell\|^2$ is the regularization term ($w_\ell = u_\ell - \Theta^T v_\ell$) controlling the task relatedness among m tasks, and α is the pre-specified non-negative parameter.

The optimization problem in Eq. (3.5) is non-convex due to its orthonormal constraint and the regularization term in terms of u_ℓ, v_ℓ, and Θ. We present an improved ASO formulation (called iASO) given by:

$$\begin{array}{ll} (\mathbf{F}_0) \min_{\{u_\ell, v_\ell\}, \Theta} & \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) + g_\ell(u_\ell, v_\ell, \Theta) \right), \\ \text{subject to} & \Theta\Theta^T = I_{h \times h}, \end{array} \quad (3.6)$$

where $g_\ell(u_\ell, v_\ell, \Theta)$ is the regularization function defined as:

$$\begin{aligned} g_\ell(u_\ell, v_\ell, \Theta) &= \alpha \|w_\ell\|^2 + \beta \|u_\ell\|^2 \\ &= \alpha \|u_\ell - \Theta^T v_\ell\|^2 + \beta \|u_\ell\|^2. \end{aligned} \quad (3.7)$$

The regularization function in Eq. (3.7) controls the task relatedness (via the first component) as well as the complexity of the predictor functions (via the second component) as commonly used in traditional regularized risk minimization formulation for supervised learning. Note that α and β are pre-specified coefficients, indicating the importance of the corresponding regularization component. For simplicity, we use the same α and β parameters for all tasks. However, the discussion below can be easily extended to the case where α and β are different for different tasks.

The iASO formulation ($\mathbf{F}_0$ in Eq. (3.6)) subsumes several multi-task learning algorithms as special cases: it reduces to the ASO algorithm in Eq. (3.5) by setting $\beta = 0$ in Eq. (3.7); and it reduces to m independent quadratic programs (QP) by setting $\alpha = 0$. It is worth noting that $\mathbf{F}_0$ is non-convex. In the next section, we convert F_0 into a (relaxed) convex formulation, which admits a globally optimal solution.

3.3 A Convex Multi-Task Learning Formulation

In this section, we consider a convex relaxation of the non-convex problem $\mathbf{F}_0$ (iASO) in Eq. (3.6). The optimal $\{v_\ell^*\}_{\ell=1}^m$ to Eq. (3.6) can be expressed in the form of a function on Θ and $\{u_\ell\}_{\ell=1}^m$. It can be verified that

$$v_\ell^* = \Theta u_\ell = \arg\min_{v_\ell} g_\ell(u_\ell, v_\ell, \Theta),\ \ell \in \mathbb{N}_m. \tag{3.8}$$

Let $U = [u_1, \cdots, u_m] \in \mathbb{R}^{d\times m}$ and $V = [v_1, \cdots, v_m] \in \mathbb{R}^{h\times m}$. From Eq. (3.8), the optimal V^* to Eq. (3.6) is given by $V^* = \Theta U$. Therefore we denote

$$\begin{aligned} G_0(U,\Theta) &= \min_V \sum_{\ell=1}^m g_\ell(u_\ell, v_\ell, \Theta) \\ &= \sum_{\ell=1}^m \alpha\left(\|u_\ell - \Theta^T\Theta u_\ell\|^2\right) + \beta\|u_\ell\|^2 \\ &= \alpha\ \text{tr}\left(U^T\left((1+\eta)I - \Theta^T\Theta\right)U\right), \end{aligned} \tag{3.9}$$

where the parameter η is defined as

$$\eta = \frac{\beta}{\alpha} > 0. \tag{3.10}$$

Moreover, it can be verified that the following equality holds

$$(1+\eta)I - \Theta^T\Theta = \eta\,(1+\eta)\left(\eta I + \Theta^T\Theta\right)^{-1}. \tag{3.11}$$

We can then reformulate $G_0(U,\Theta)$ in Eq. (3.9) into an equivalent form given by

$$G_1(U,\Theta) = \alpha\,\eta\,(1+\eta)\ \text{tr}\left(U^T\left(\eta I + \Theta^T\Theta\right)^{-1}U\right). \tag{3.12}$$

Since the loss term in Eq. (3.6) is independent of the optimization variables $\{v_\ell\}_{\ell=1}^m$, $\mathbf{F}_0$ can be equivalently transformed into the following optimization problem $\mathbf{F}_1$ with optimization variables Θ and U:

$$\begin{aligned} (\mathbf{F}_1)\ \min_{\{u_\ell\},\Theta} \quad & \sum_{\ell=1}^m \left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell)\right) + G_1(U,\Theta) \\ \text{subject to} \quad & \Theta\Theta^T = I_{h\times h}. \end{aligned} \tag{3.13}$$

where $G_1(U,\Theta)$ is defined in Eq. (3.12).

Convex Relaxation

The orthonormality constraints in Eq. (3.13) is non-convex, so is the optimization problem $\mathbf{F}_1$. We propose to convert $\mathbf{F}_1$ into a convex formulation by relaxing its feasible domain into a convex set.

Let the set $\mathcal{M}_e$ be defined as:

$$\mathcal{M}_e = \left\{M_e \mid M_e = \Theta^T\Theta,\ \Theta\Theta^T = I,\ \Theta \in \mathbb{R}^{h\times d}\right\}. \tag{3.14}$$

It has been shown in [59] that the convex hull [60] of $\mathcal{M}_e$ can be precisely expressed as the convex set $\mathcal{M}_c$ given by

$$\mathcal{M}_c = \left\{M_c \mid \mathrm{tr}(M_c) = h,\ M_c \preceq I,\ M_c \in \mathbb{S}_+^d\right\}, \tag{3.15}$$

and each element in $\mathcal{M}_e$ is referred to as an extreme point of $\mathcal{M}_c$. Since $\mathcal{M}_c$ consists of all convex combinations of the elements in $\mathcal{M}_e$, $\mathcal{M}_c$ is the smallest convex set that contains $\mathcal{M}_e$, and $\mathcal{M}_e \subseteq \mathcal{M}_c$.

To convert the non-convex problem **F**$_1$ into a convex formulation, we replace $\Theta^T\Theta$ with M in Eq. (3.13), and naturally relax its feasible domain into a convex set based on the relationship between $\mathcal{M}_e$ and $\mathcal{M}_c$ presented above; this results in an optimization problem **F**$_2$ (called rASO) as:

$$\begin{array}{ll} (\mathbf{F}_2)\ \min\limits_{\{u_\ell\},M} & \displaystyle\sum_{\ell=1}^{m}\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell)\right) + G_2(U,M) \\ \text{subject to} & \mathrm{tr}(M) = h,\ M \preceq I,\ M \in \mathbb{S}_+^d, \end{array} \tag{3.16}$$

where $G_2(U,M)$ is defined as:

$$G_2(U,M) = \alpha\,\eta\,(1+\eta)\ \mathrm{tr}\left(U^T\left(\eta I + M\right)^{-1}U\right). \tag{3.17}$$

It follows from [61, Theorem 3.1] that the regularization term $G_2(U,M)$ is jointly convex in U and M and the optimization problem **F**$_2$ is convex. Note that **F**$_2$ is a convex relaxation of **F**$_1$ as the optimal M to **F**$_2$ is not guaranteed to occur at the extreme points of $\mathcal{M}_c$. The optimal Θ to **F**$_1$ can be approximated using the first h eigenvectors (corresponding to the largest h eigenvalues) of the optimal M computed from **F**$_2$.

The SDP Formulation

The optimization problem **F**$_2$ can be readily reformulated into an equivalent semi-definite program (SDP) [60]. We add slack variables $\{t_\ell\}_{\ell=1}^m$ and enforce

$$u_\ell^T(\eta I + M)^{-1}u_\ell \le t_\ell,\ \forall \ell \in \mathbb{N}_m. \tag{3.18}$$

It follows from the Schur complement Lemma [62] that we can rewrite $\mathbf{F}_2$ as:

$$
\begin{aligned}
(\mathbf{F}_3) \min_{\{u_\ell, t_\ell\}, M} \quad & \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) \right) + \alpha\eta(1+\eta) \sum_{\ell=1}^{m} t_\ell \\
\text{subject to} \quad & \begin{pmatrix} \eta I + M & u_\ell \\ u_\ell^T & t_\ell \end{pmatrix} \succeq 0, \ \forall \ell \in \mathbb{N}_m, \\
& \text{tr}(M) = h, \ M \preceq I, \ M \in \mathbb{S}_+^d. \qquad (3.19)
\end{aligned}
$$

Given that the loss function L is convex, the optimization problem $\mathbf{F}_3$ is convex. However, it is not scalable to large scale data sets due to its positive semidefinite constraints. If L is the SVM hinge loss, $\mathbf{F}_3$ is an SDP. Note that many off-the-shelf optimization solvers such as SeDuMi[2] can be used for solving SDP, which can only handle several hundreds of optimization variables.

3.4 Accelerated Projected Gradient Algorithm

In this section, we propose to apply the accelerated projected gradient (APG) algorithm [45] for solving an equivalent counterpart of $\mathbf{F}_3$ in Eq. (3.19), i.e., $\mathbf{F}_2$ in Eq. (3.16); we also develop efficient algorithms for solving the key sub-problems involved in each iteration of the APG algorithm. We present a concrete example to illustrate the APG algorithm for solving rASO in Eq. (15) with the Hinge loss function in the supplementary file.

The Main Algorithm

For notational simplicity, we denote the convex optimization problem in Eq. (3.16) as

$$
\begin{aligned}
\min_{Z} \quad & f(Z) + g(Z) \\
\text{subject to} \quad & Z \in \mathcal{C}, \qquad (3.20)
\end{aligned}
$$

where Z symbolically represents the optimization variables U_Z and M_Z as

$$
Z = \begin{bmatrix} U_Z \\ M_Z \end{bmatrix}, \ U_Z \in \mathbb{R}^{d \times m}, \ M_Z \in \mathbb{R}^{d \times d},
$$

$\mathcal{C}$ is a closed and convex domain set defined as

$$
\mathcal{C} = \left\{ Z \mid U_Z \in \mathbb{R}^{d \times m}, \text{tr}(M_Z) = h, M_Z \preceq I, M_Z \in \mathbb{S}_+^d \right\},
$$

$f(Z)$ and $g(Z)$ respectively denote the smooth and non-smooth components of the objective function in Eq. (3.16). Note that in the following presentation we assume that the smooth function $f(Z)$

[2] http://sedumi.ie.lehigh.edu/

has a Lipschitz continuous gradient $\mathcal{L}_f$ [63] as:

$$\|\nabla f(Z_x) - \nabla f(Z_y)\|_F \leq \mathcal{L}_f \|Z_x - Z_y\|_F \tag{3.21}$$

for any pair of $Z_x, Z_y \in \mathcal{C}$. Since the regularization term in Eq. (3.16) is smooth, the component $g(Z)$ in Eq. (3.20) vanishes if the loss function L is smooth.

Given a non-negative parameter γ, we define a construction as

$$f_\gamma(Z, S) = f(S) + \langle \nabla f(S), Z - S \rangle + \frac{\gamma}{2}\|Z - S\|_F^2.$$

Moreover, we define

$$F_\gamma(Z, S) = f_\gamma(Z, S) + g(Z).$$

It can be easily verified that $F_\gamma(Z, S)$ is strictly convex with respect to Z. For any given S and γ, the unique minimizer $\hat{Z}$ to $F_\gamma(Z, S)$ can be obtained via

$$\begin{aligned} \hat{Z} &= \arg\min_{Z \in \mathcal{C}} F_\gamma(Z, S) \\ &= \arg\min_{Z \in \mathcal{C}} \left(\frac{\gamma}{2} \left\| Z - \left(S - \frac{1}{\gamma} \nabla f(S) \right) \right\|_F^2 + g(Z) \right). \end{aligned} \tag{3.22}$$

Denote the objective function in Eq. (3.20) by $F(Z)$ as

$$F(Z) = f(Z) + g(Z).$$

If the non-negative parameter γ satisfies the inequality

$$F(\hat{Z}) \leq F_\gamma(\hat{Z}, S), \tag{3.23}$$

we say that γ is appropriate [57] for $\hat{Z}$, where $\hat{Z}$ is the minimizer obtained via Eq. (3.22).

To solve the optimization problem in Eq. (3.20), the APG algorithm constructs a solution point sequence $\{Z_k\}$ and a searching point sequence $\{S_k\}$, where each Z_k is updated from S_k via Eq. (3.22). The pseudo-code of the APG algorithm is presented in Algorithm 8. Using standard techniques in [45,57], we can show that Algorithm 8 attains the convergence rate of $\mathcal{O}(1/k^2)$, where k denotes the number of iterations. Note that the APG algorithm belongs to the category of the first-order methods and its convergence rate is optimal among all first-order methods [57].

Efficient Algorithms

The APG algorithm requires to solve the constrained optimization problem in Eq. (3.22) in each of its iterations. In Eq. (3.22), the objective function consists of a smooth component and a non-smooth component (if the employed loss function is non-smooth); we propose efficient algorithms for solving this composite optimization problem.

```
Input: $Z_0$, $\gamma_0 \in \mathbb{R}$, and max-iter.
Output: $Z$.
Set $Z_1 = Z_0$, $t_{-1} = 0$, and $t_0 = 1$.
for $i = 1, 2, \cdots$, max-iter do
    Compute $\alpha_i = (t_{i-2} - 1)/t_{i-1}$.
    Compute $S = (1 + \alpha_i)Z_i - \alpha_i Z_{i-1}$.
    while (true)
        Compute $\hat{Z}$ via Eq. (3.22).
        if $F(\hat{Z}) \leq F_\gamma(\hat{Z}, S)$ then exit the loop
            else update $\gamma_i = \gamma_i \times 2$.
        end-if
    end-while
    Update $Z_{i+1} = \hat{Z}$ and $\gamma_{i+1} = \gamma_i$.
    if stopping criteria satisfied then exit the loop.
    Update $t_i = \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$.
end-for
Set $Z = Z_{i+1}$.
```

Algorithm 3: Accelerated Projected Gradient Algorithm for Multi-Task Learning

Smooth Loss Function

If the loss function L in Eq. (3.16) is smooth, the non-smooth component $g(Z)$ in the symbolical form of Eq. (3.20) vanishes. We can express $f(Z)$ and $g(Z)$ as

$$\begin{aligned} f(Z) &= \sum_{\ell=1}^{m} \sum_{i=1}^{n_\ell} \frac{1}{n_\ell} L(u_{z\ell}^T x_i^\ell, y_i^\ell) + c \,\text{tr}\left(U_Z^T \left(\eta I + M_Z\right)^{-1} U_Z \right) \\ g(Z) &= 0, \end{aligned} \tag{3.24}$$

where $U_Z = [u_{z1}, \cdots, u_{zm}]$ and $c = \alpha\eta(1+\eta)$. Note that the commonly used smooth loss functions include Least Squares Loss, Logistic Regress Loss, and Huber's Robust Loss.

In the setting of employing the smooth loss functions in Eq. (3.16), the optimization problem in Eq. (3.22) can be correspondingly expressed as

$$\begin{aligned} \min_{U_Z, M_Z} \quad & \left\| U_Z - \hat{U}_S \right\|_F^2 + \left\| M_Z - \hat{M}_S \right\|_F^2 \\ \text{subject to} \quad & \text{tr}(M_Z) = h, \ M_Z \preceq I, \ M_Z \in \mathbb{S}_+^d, \end{aligned} \tag{3.25}$$

where $\hat{U}_S = U_S - \frac{1}{\gamma}\nabla_{U_S} f(S)$ and $\hat{M}_S = M_S - \frac{1}{\gamma}\nabla_{M_S} f(S)$. Note that S symbolically represents

$$S = \begin{bmatrix} U_S \\ M_S \end{bmatrix}, \ U_S \in \mathbb{R}^{d \times m}, \ M_S \in \mathbb{R}^{d \times d},$$

$\nabla_{U_S} f(S)$ and $\nabla_{M_S} f(S)$ denote the derivative of $f(S)$ with respect to U_S and M_S, respectively. It can be verified that the optimal U_Z and M_Z to Eq. (3.25) can be obtained by solving two optimization problems independently as below.

Computation of U_Z The optimal U_Z to Eq. (3.25) can be obtained by solving

$$\min_{U_Z} \quad \left\| U_Z - \hat{U}_S \right\|_F^2. \tag{3.26}$$

Obviously the optimal U_Z to Eq. (3.26) is given by $\hat{U}_S$.

Computation of M_Z The optimal M_Z to Eq. (3.25) can be obtained by solving

$$\begin{aligned} \min_{M_Z} \quad & \left\| M_Z - \hat{M}_S \right\|_F^2 \\ \text{subject to} \quad & \text{tr}(M_Z) = h,\ M_Z \preceq I,\ M_Z \in \mathbb{S}_+^d, \end{aligned} \tag{3.27}$$

where $\hat{M}_S$ is symmetric but may not be positive semi-definite (PSD). The optimal M_Z to Eq. (3.27) can be computed via solving a simple convex projection problem, as summarized in Theorem 3.4.1. Before presenting Theorem 3.4.1, we present a lemma, which is important for the analysis in Theorem 3.4.1.

Lemma 3.4.1. *Given an arbitrary diagonal matrix $E =$ diag$(e_1, \cdots, e_d) \in \mathbb{R}^{d\times d}$, let the optimization problem A be defined as*

$$\begin{aligned} \min_{T} \quad & \|T - E\|_F^2 \\ \textit{subject to} \quad & \textit{tr}(T) = h,\ 0 \preceq T \preceq I, \end{aligned}$$

and let the optimization problem B be defined as

$$\begin{aligned} \min_{\hat{E}} \quad & \|\hat{E} - E\|_F^2 \\ \textit{subject to} \quad & \textit{tr}(\hat{E}) = h,\ \hat{E} = \textit{diag}(\hat{e}_1, \cdots, \hat{e}_d),\ 0 \le \hat{e}_i \le 1. \end{aligned}$$

Denote the optimal objective value of problem A by O_A, and the optimal objective value of problem B by O_B. Then

$$O_A = O_B.$$

Proof. Since any feasible solution point of problem B must be feasible for problem A, we have $O_A \le O_B$. Let T^* be the optimal solution to problem A, and $diag(T^*)$ be the diagonal matrix obtained by setting the off-diagonal entries of T^* as zeros. It follows that

$$O_A = \|T^* - E\|_F^2 \ge \|diag(T^*) - E\|_F^2.$$

It can be easily verified that

$$\text{tr}(diag(T^*)) = h,\ 0 \preceq diag(T^*) \preceq I, \tag{3.28}$$

and $diag(T^*)$ is feasible in problem B. Therefore

$$\|diag(T^*) - E\|_F^2 \geq O_B,\ O_A \geq O_B.$$

This completes the proof of this lemma. □

We now show how to compute the optimal M_Z to Eq. (3.27).

Theorem 3.4.1. *Given an arbitrary symmetric matrix $\hat{M}_S \in \mathbb{R}^{d\times d}$ in Eq. (3.27), let $\hat{M}_S = P\hat{\Sigma}P^T$ be its eigendecomposition, where $P \in \mathbb{R}^{d\times d}$ is orthogonal, and $\hat{\Sigma} =$ diag$(\hat{\sigma}_1, \cdots, \hat{\sigma}_d) \in \mathbb{R}^{d\times d}$ is diagonal with the eigenvalues on its main diagonal. Let $\Sigma^* =$ diag$(\sigma_1^*, \cdots, \sigma_d^*) \in \mathbb{R}^{d\times d}$, where $\{\sigma_i^*\}_{i=1}^d$ is the optimal solution to the following optimization problem:*

$$\begin{aligned} \min_{\{\sigma_i\}} \quad & \sum_{i=1}^{d} (\sigma_i - \hat{\sigma}_i)^2 \\ \textit{subject to} \quad & \sum_{i=1}^{d} \sigma_i = h,\ 0 \leq \sigma_i \leq 1,\ i = 1, \cdots, d. \end{aligned} \tag{3.29}$$

Then the global minimizer to Eq. (3.27) is given by $M^* = P\Sigma^*P^T$.

Proof. For arbitrary M_Z feasible in Eq. (3.27), we denote its eigendecomposition by $M_Z = Q\Lambda Q^T$, where $Q \in \mathbb{R}^{d\times d}$ is orthogonal, $\Lambda =$ diag$(\lambda_1, \cdots, \lambda_d) \in \mathbb{R}^{d\times d}$ is diagonal with the eigenvalues on its main diagonal. Since the orthogonal transformation does not change the Euclidean distance, the optimization problem in Eq. (3.27) is equivalent to

$$\begin{aligned} \min_{\Lambda, Q} \quad & \left\| P^T Q \Lambda Q^T P - \hat{\Sigma} \right\|_F^2 \\ \text{subject to} \quad & \text{tr}(\Lambda) = h,\ \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d),\ 0 \leq \lambda_i \leq 1 \\ & Q^T Q = Q Q^T = I_d, \end{aligned} \tag{3.30}$$

where Λ and Q are two separate optimization variables. From Lemma 3.4.1, we have that Eq. (3.29) and Eq. (3.30) admit the same optimal objective value. It can be easily verified that the solution pair $\{\Lambda = \Sigma^*, Q = P\}$ is feasible in Eq. (3.30) and attain the optimal objective value. Since the problem in Eq. (3.30) is strictly convex, $M^* = P\Sigma^*P^T$ is the unique global minimizer to Eq. (3.27). This completes the proof. □

Non-Smooth Loss Function

If the loss function L in Eq. (3.16) is non-smooth, the smooth component $f(Z)$ in Eq. (3.20) can be expressed as

$$f(Z) = c\ \text{tr}\left(U_Z^T \left(\eta I + M_Z\right)^{-1} U_Z \right), \tag{3.31}$$

where $c = \alpha\eta(1+\eta)$; the non-smooth component $g(Z)$ can be expressed as

$$g(Z) = \sum_{\ell=1}^{m} \sum_{i=1}^{n_\ell} \frac{1}{n_\ell} L({u_{z\ell}}^T x_i^\ell, y_i^\ell), \tag{3.32}$$

where $U_Z = [u_{z1}, \cdots, u_{zm}]$. Since $g(Z)$ is independent of the variable M_Z in Eq. (3.32), for clear specification, we denote $g(Z)$ by $g(U_Z)$ in the following presentation. Note that the commonly used non-smooth loss function includes SVM Hinge Loss.

In the setting of employing non-smooth loss functions, the optimization problem in Eq. (3.22) can be correspondingly expressed as

$$\begin{aligned} \min_{U_Z, M_Z} \quad & \left\| U_Z - \hat{U}_S \right\|_F^2 + \left\| M_Z - \hat{M}_S \right\|_F^2 + \hat{\gamma} g(U_Z) \\ \text{subject to} \quad & \text{tr}(M_Z) = h,\ M_Z \preceq I,\ M_Z \in \mathbb{S}_+^d, \end{aligned} \tag{3.33}$$

where $\hat{U}_S = U_S - \frac{1}{\gamma}\nabla_{U_S} f(S)$, $\hat{M}_S = M_S - \frac{1}{\gamma}\nabla_{M_S} f(S)$, and $\hat{\gamma} = \frac{2}{\gamma}$. The optimization problem in Eq. (3.33) is non-smooth convex with two decoupled optimization variables U_Z and M_Z. Similarly, the optimal U_Z and M_Z to Eq. (3.33) can be obtained by solving two convex optimization problems independently.

Computation of U_Z The optimal U_Z to Eq. (3.33) can be obtained by solving

$$\min_{U_Z} \quad \left\| U_Z - \hat{U}_S \right\|_F^2 + \hat{\gamma} g(U_Z). \tag{3.34}$$

The optimization problem in Eq. (3.34) can be solved using different approaches depending on the specific structures of the non-smooth component $g(U_Z)$. When SVM Hinge Loss is employed, Eq. (3.34) can be reformulated as a set of QP problems and solved via many sophisticated optimization solvers.

Computation of M_Z The optimal M_Z to Eq. (3.33) can be obtained by solving

$$\begin{aligned} \min_{M_Z} \quad & \left\| M_Z - \hat{M}_S \right\|_F^2 \\ \text{subject to} \quad & \text{tr}(M_Z) = h,\ M_Z \preceq I,\ M_Z \in \mathbb{S}_+^d. \end{aligned} \tag{3.35}$$

Similar to the case with the smooth loss function, the optimal M_Z to Eq. (3.35) can be obtained by solving a simple convex optimization problem following the results in Theorem 3.4.1.

Discussion

We discuss the main computational cost of the APG algorithm for solving Eq. (3.16) in the setting of using the smooth loss functions and the non-smooth loss functions, respectively.

Using the smooth loss functions The main computational procedures in each iteration of the APG algorithm include the computation of Eq. (3.26) and Eq. (3.27). The optimal solution to Eq. (3.26) can be trivially obtained; the optimal solution to Eq. (3.27) can be obtained via computing eigen-decomposition of a symmetric matrix of size $d \times d$ and solving a simple convex optimization in Eq. (3.29) as presented in Theorem 3.4.1.

Using the non-smooth loss functions The main computational procedures in each iteration of the APG algorithm include the computation of Eq. (3.34) and Eq. (3.35). The optimal solution to Eq. (3.34) can be obtained via solving a set of QP problems; each QP problem admits a sparse hessian matrix (an identity matrix of size $d \times d$). The optimal solution to Eq. (3.35) can be similarly obtained via computing eigen-decomposition of a symmetric matrix of size $d \times d$ and solving a simple convex optimization in Eq. (3.29).

Clearly the computation complexity of the APG algorithm for solving Eq. (3.16) is primarily dependent on the feature dimensionality in the data. It could be advantageous to apply APG for the setting of using smooth convex loss functions, because the optimization of U_Z is trivial and the computation of QP problems can be avoided (compared to the setting of using non-smooth convex loss functions).

3.5 Convex Alternating Structure Optimization Algorithm

In this section, we propose a convex alternating structure optimization (called CASO) algorithm to solve the optimization problem $\mathbf{F}_2$ in Eq. (3.16). In essence, CASO is similar to the block coordinate descent (BCD) method [58], in which the optimization variables are optimized alternatively with the rest of the optimization variables fixed. The pseudo-codes of the CASO algorithm are presented in Algorithm 4, and the detailed computational procedures are presented below. We present a concrete example to illustrate the CASO algorithm for solving rASO in Eq. (15) with the Hinge loss function in the supplementary file.

Computation of U for a Given M

For a fixed M, the optimal U can be computed by solving the following problem:

$$\min_{\{u_\ell\}} \quad \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) + \hat{g}(u_\ell) \right), \tag{3.36}$$

where the regularization term $\hat{g}(u_\ell)$ is given by

$$\hat{g}(u_\ell) = \alpha \, \eta \, (1+\eta) \, \mathrm{tr} \left(u_\ell^T \left(\eta I + M \right)^{-1} u_\ell \right), \; \ell \in \mathbb{N}_m.$$

```
Input: $\{(x_i^\ell, y_i^\ell)\}$, $i \in \mathbb{N}_{n_\ell}$, $\ell \in \mathbb{N}_m$, $h \in \mathbb{N}$.
Output: $U$, $V$, and $\Theta$.
Parameter: $\alpha$ and $\beta$.
Initialize $M$ subject to the constraints in Eq. (3.16).
repeat
    Update $U$ via Eq. (3.36).
    Compute the SVD of $U$ as $U = P_1 \Sigma P_2^T$.
    Compute $\{\gamma_i^*\}_{i=1}^q$ via Eq. (3.39).
    Update $M$ as $M = P_1 \text{diag}(\gamma_1^*, \cdots, \gamma_q^*) P_1^T$.
until convergence criterion is satisfied.
Construct $\Theta$ using the top $h$ eigenvectors of $M$.
Construct $V$ as $V = \Theta U$.
Return $U$, $V$ and $\Theta$.
```

Algorithm 4: CASO for Multi-Task Learning

Given any convex loss function L, the objective function in Eq. (3.36) is strictly convex, and hence the corresponding optimization problem admits a unique minimizer. Note that the number of optimization variables in Eq. (3.36) depends on the sample size and the feature dimensionality simultaneously, and the hessian matrix is not sparse in general. Instead of solving Eq. (3.36) directly, we propose to solve its equivalent dual form, in which the number of optimization variables is independent of the dimensionality. In Section 4.6, we present an example on the conversion of Eq. (3.36) (with hinge loss) to its dual form.

Computation of M for a Given U

For a fixed U, the optimal M can be computed by solving the following problem:

$$\begin{aligned} \min_{M} \quad & \text{tr}\left(U^T (\eta I + M)^{-1} U\right) \\ \text{subject to} \quad & \text{tr}(M) = h, M \preceq I, M \in \mathbb{S}_+^d. \end{aligned} \tag{3.37}$$

This problem can be recast into an SDP problem, which is computationally expensive to solve. We propose an efficient approach to solve the optimization problem in Eq. (3.37); its optimal solution can be obtained via solving a simple eigenvalue optimization problem.

Efficient Computation of Eq. (3.37) For any $U \in \mathbb{R}^{d \times m}$ in Eq. (3.37), let $U = P_1 \Sigma P_2^T$ be its SVD [62], where $P_1 \in \mathbb{R}^{d \times d}$, $P_2 \in \mathbb{R}^{m \times m}$ are orthogonal, and $\Sigma \in \mathbb{R}^{d \times m}$ has q nonzero singular values on its main diagonal ($q \leq m \leq d$). We denote

$$\begin{aligned} & \Sigma = \left[\text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m); \mathbf{0}_{(d-m)\times m}\right] \in \mathbb{R}^{d \times m}, \\ & \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q > 0 = \sigma_{q+1} = \cdots = \sigma_m. \end{aligned} \tag{3.38}$$

Note that since the value of h controls the size of the shared low-dimensional structure, we focus on the setting of $h \leq q \leq m \leq d$. We show that the optimal M to Eq. (3.37) can be obtained via solving the following convex optimization problem [60]:

$$\begin{aligned} \min_{\{\gamma_i\}_{i=1}^{q}} \quad & \sum_{i=1}^{q} \frac{\sigma_i^2}{\eta + \gamma_i} \\ \text{subject to} \quad & \sum_{i=1}^{q} \gamma_i = h, \; 0 \leq \gamma_i \leq 1. \end{aligned} \tag{3.39}$$

We summarize an important property of the optimal solution to Eq. (3.39) in the following lemma.

Lemma 3.5.1. *The optimal $\{\gamma_i^*\}_{i=1}^{q}$ to Eq. (3.39) satisfy $\gamma_1^* \geq \gamma_2^* \cdots \geq \gamma_q^*$.*

Proof. Prove by contradiction. For any $\sigma_i > \sigma_{i+1}$, we assume $\gamma_i^* < \gamma_{i+1}^*$. We can construct another feasible solution by switching the positions of γ_i^* and γ_{i+1}^*, and attain a smaller objective value in Eq. (3.39), leading to a contradiction. This completes the proof of this lemma. □

Note that the optimization problem in Eq. (3.39) can be solved via many existing algorithms such as the projected gradient descent method [60]. An immediate and obvious consequence from the results of Lemma 3.5.1 is

$$\frac{1}{\eta + \gamma_1^*} \leq \frac{1}{\eta + \gamma_2^*} \leq \cdots \leq \frac{1}{\eta + \gamma_q^*}. \tag{3.40}$$

Before presenting the efficient approach for solving Eq. (3.37), we first present the following lemma, which will be useful for our following analysis.

Lemma 3.5.2. *For any matrix $Z \in \mathbb{S}_+^d$, let $Z = \hat{U}\hat{\Sigma}_z\hat{U}^T$ be its SVD, where $\hat{U} \in \mathbb{R}^{d\times d}$ is orthogonal, $\hat{\Sigma}_z =$ diag$(\hat{\sigma}_1, \cdots, \hat{\sigma}_d)$, and $\hat{\sigma}_1 \geq \cdots \geq \hat{\sigma}_d \geq 0$. Let $\{Z_i\}_{i=1}^{d}$ be the diagonal entries of Z, and $\Pi = \{\pi_1, \cdots, \pi_p\} \subseteq \mathbb{N}_d$ be any integer subset with p $(p \leq d)$ distinct elements. Then $\sum_{i=1}^{p} Z_{\pi_i} \leq \sum_{j=1}^{p} \hat{\sigma}_j$.*

Proof. Denote the i-th row-vector of $\hat{U} \in \mathbb{R}^{d\times d}$ by $\hat{U}_i = [\hat{u}_{i1}, \cdots, \hat{u}_{id}]$. For any integer subset $\Pi = \{\pi_1, \cdots, \pi_p\}$, we have

$$0 \leq \sum_{k=1}^{p} \hat{u}_{\pi_k j}^2 \leq 1, \; \sum_{j=1}^{d} \hat{u}_{\pi_k j}^2 = 1, \; \forall j \in \mathbb{N}_d, \; \forall k \in \mathbb{N}_p.$$

The i-th diagonal entry of Z can be expressed as $Z_i = \sum_{j=1}^{d} \hat{\sigma}_j \hat{u}_{ij}^2$. It follows that

$$\begin{aligned} \sum_{i=1}^{p} Z_{\pi_i} &= \sum_{j=1}^{d} \left(\hat{\sigma}_j \hat{u}_{\pi_1 j}^2 + \cdots + \hat{\sigma}_j \hat{u}_{\pi_p j}^2\right) \\ &= \sum_{j=1}^{d}\sum_{k=1}^{p} \left(\hat{\sigma}_j \hat{u}_{\pi_k j}^2\right) = \sum_{j=1}^{d} \left(\hat{\sigma}_j \sum_{k=1}^{p} \hat{u}_{\pi_k j}^2\right) \leq \sum_{j=1}^{p} \hat{\sigma}_j, \end{aligned}$$

where the last equality (the maximum) above is attained when the set $\{\hat{u}^2_{\pi_1 j}, \cdots, \hat{u}^2_{\pi_p j}\}$ $(\forall j \in \mathbb{N}_d)$ has only one non-zero element of value one or $p = d$. This completes the proof of this lemma. □

We summarize the main result of the efficient approach for solving Eq. (3.37) in the following theorem.

Theorem 3.5.1. *Let $\{\lambda^*_i\}_{i=1}^q$ be optimal to Eq. (3.39), and denote $\Lambda^* = \text{diag}(\lambda^*_1, \cdots, \lambda^*_q, 0) \in \mathbb{R}^{d\times d}$. Let $P_1 \in \mathbb{R}^{d\times d}$ be orthogonal consisting of the left singular vectors of U. Then $M^* = P_1 \Lambda^* P_1^T$ is an optimal solution to Eq. (3.37). Moreover, the problem in Eq. (3.39) attains the same optimal objective value as the one in Eq. (3.37).*

Proof. For any feasible M in Eq. (3.37), let $M = Q\Lambda Q^T$ be its SVD, where $Q \in \mathbb{R}^{d\times d}$ is orthogonal, $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d)$, and $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. The problem in Eq. (3.37) can be rewritten as:

$$\begin{aligned} \min_{Q,\Lambda} \quad & \text{tr}\left((\eta I + \Lambda)^{-1} Q^T P_1 \Sigma \Sigma^T P_1^T Q \right) \\ \text{subject to} \quad & QQ^T = Q^T Q = I,\ \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d), \\ & \sum_{i=1}^d \lambda_i = h,\ 1 \geq \lambda_1 \geq \cdots \geq \lambda_d \geq 0, \end{aligned} \tag{3.41}$$

where Σ is defined in Eq. (3.38). Note that the reformulated problem in Eq. (3.41) is equivalent to the one in Eq. (3.37) and has two separate optimization variables Q and Λ.

We show that the optimization variable Q can be factored out from Eq. (3.41), and the optimal Q^* can be obtained analytically. Let $D = Q^T P_1 \Sigma \Sigma^T P_1^T Q$ and denote its diagonal entries by $\{D_i\}_{i=1}^d$. It follows from Eq. (3.38) that D is a positive semidefinite matrix with non-zero singular values $\{\sigma_i^2\}_{i=1}^q$. Given any feasible Λ in Eq. (3.41), we have

$$\min_{Q^TQ=QQ^T=I} \text{tr}\left((\eta I + \Lambda)^{-1} Q^T P_1 \Sigma \Sigma^T P_1^T Q \right) = \min_{D \in \mathbb{S}^d_+ : D \sim \Sigma\Sigma^T} \sum_{i=1}^d \frac{D_i}{\eta + \lambda_i}, \tag{3.42}$$

where $D \sim \Sigma\Sigma^T$ indicates that the eigenvalues of D are given by the diagonal elements of $\Sigma\Sigma^T$, and the equality above means that these two problems attain the same optimal objective value. Following the non-decreasing order of $1/(\eta + \lambda_i)$ $(i \in \mathbb{N}_q)$ in Eq. (3.40) and $\sum_{i=1}^p D_{\pi_i} \leq \sum_{j=1}^p \sigma_j^2$ for any integer subset $\{\pi_i\}_{i=1}^p$ (Lemma 3.5.2), we can verify that the optimal objective value to Eq. (3.42) is given by

$$\begin{aligned} \min_{D \in \mathbb{S}^d_+ : D \sim \Sigma\Sigma^T} \sum_{i=1}^d \frac{D_i}{\eta + \lambda_i} &= \sum_{i=1}^q \frac{\sigma_i^2}{\eta + \lambda_i} + \sum_{i=q+1}^d \frac{0}{\eta + \lambda_i} \\ &= \sum_{i=1}^q \frac{\sigma_i^2}{\eta + \lambda_i}, \end{aligned} \tag{3.43}$$

where this optimum can be attained when $Q^T P_1 = I$ [62] and $D = \Sigma\Sigma^T$. It follows from Eq. (3.43) that the optimal $\{\lambda_i^*\}_{i=1}^d$ to Eq. (3.39) satisfy $\lambda_{q+1}^* = \cdots = \lambda_d^* = 0$.

In summary, the optimal objective value to Eq. (3.41) or equivalently Eq. (3.37) can be obtained via solving Eq. (3.43) subject to the constraints on $\{\lambda_i\}$ or equivalently Eq. (3.39). Since Eq. (3.42) is minimized when $Q = P_1$, we conclude that $M^* = P_1 \Lambda^* P_1^T$ is optimal to Eq. (3.37). This completes the proof. □

Note that the optimization problem (not strictly convex) in Eq. (3.37) may have multiple global minimizers yet with the same objective value, while the formulation in Eq. (3.39) can find one of those global minimizers.

Discussion

The alternating optimization procedure employed in Algorithm 4 (CASO) is widely used for for solving many optimization problems efficiently. However, such a procedure does not generally guarantee the global convergence. We summarize the global convergence property of CASO algorithm in the following theorem. We omit the detailed proof for Theorem 3.5.2, as the proof follows similar arguments in [53, 61].

Theorem 3.5.2. *Algorithm 4 converges to the global minimizer of the optimization problem* $\boldsymbol{F}_2$ *in Eq. (3.16).*

The CASO algorithm computes the optimal solution to Eq. (3.16) by iteratively solving the dual form of Eq. (3.36) and Eq. (3.37). For the dual form of Eq. (3.36), the number of optimization variables depends on the sample size; for Eq. (3.37), the optimal solution can be obtained via computing the economic SVD of a matrix of size $d \times m$ (in general $d \gg m)$ and solving an simple singular value projection problem in Eq. (3.39). Therefore, the computation complexity of the CASO algorithm for solving Eq. (3.16) mainly depends on the sample size of the data.

3.6 Computation of an Optimal Solution to iASO

Recall that rASO in Eq. (3.16) is a convex relaxation of iASO in Eq. (3.6). In this section, we present a theoretical condition under which a globally optimal solution to iASO can be obtained via rASO.

We first present a lemma, which is the key building block of the subsequent analysis.

Lemma 3.6.1. *Let $\{\sigma_i\}_{i=1}^m$ be defined in Eq. (3.38) and $\{\gamma_i^*\}_{i=1}^q$ be optimal to Eq. (3.39). For any $h \in \mathbb{N}_q$, if $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, then $\gamma_1^* = \cdots = \gamma_h^* = 1$ and $\gamma_{h+1}^* = \cdots = \gamma_q^* = 0$.*

Proof. Prove by contradiction. Assume that $\gamma_1^* = \cdots = \gamma_h^* = 1$ and $\gamma_{h+1}^* = \cdots = \gamma_q^* = 0$ do not hold. Since $\sum_{i=1}^q \gamma_i^* = h$ and γ_i^* is non-increasing with i (Lemma 3.5.1), the assumption leads to $\gamma_h^* \neq 1$ and hence $0 < \gamma_{h+1}^* \leq \gamma_h^* < 1$. We can construct another feasible solution $\{\zeta_i^*\}_{i=1}^m$ such that $\sum_{i=1}^m \sigma_i^2/(\eta + \gamma_i^*) > \sum_{i=1}^m \sigma_i^2/(\eta + \zeta_i^*)$, thus reaching a contradiction.

Let γ_a^* be the element in $\{\gamma_i^*\}_{i=1}^q$ with the smallest index $a \in \mathbb{N}_h$, satisfying $\gamma_a^* \neq 1$. Let γ_b^* be the element in $\{\gamma_i^*\}_{i=1}^q$ with the largest index $b \in \mathbb{N}_q$, satisfying $\gamma_b^* \neq 0$. Note that it can be verified that $a \leq h$ and $h + 1 \leq b$. For any $0 < \delta < \min(1 - \gamma_a^*, \gamma_b^*)$, we can construct a feasible solution $\{\zeta_i^*\}_{i=1}^m$ to Eq. (3.39) as:

$$\zeta_i^* = \begin{cases} \gamma_i^* & i \in \mathbb{N}_q,\ i \neq a,\ i \neq b \\ \gamma_a^* + \delta & i = a \\ \gamma_b^* - \delta & i = b \end{cases}$$

such that $1 \geq \zeta_1^* \geq \cdots > \zeta_a^* > \cdots \geq \zeta_h^* > \cdots > \zeta_b^* > 0 = \cdots = 0$. Moreover, we have

$$\begin{aligned} & \left(\frac{\sigma_a^2}{\eta + \gamma_a^*} + \frac{\sigma_b^2}{\eta + \gamma_b^*}\right) - \left(\frac{\sigma_a^2}{\eta + \zeta_a^*} + \frac{\sigma_b^2}{\eta + \zeta_b^*}\right) \\ = \quad & \delta\left(\frac{\sigma_a^2}{(\eta + \gamma_a^*)(\eta + \gamma_a^* + \delta)} - \frac{\sigma_b^2}{(\eta + \gamma_b^*)(\eta + \gamma_b^* - \delta)}\right) \\ \geq \quad & \sigma_{h+1}^2\delta\left(\frac{(1 + 1/\eta)^2}{(\eta + \gamma_a^*)(\eta + \gamma_a^* + \delta)} - \frac{1}{(\eta + \gamma_b^*)(\eta + \gamma_b^* - \delta)}\right) \\ > \quad & \sigma_{h+1}^2\delta\left(\frac{(1 + 1/\eta)^2}{(\eta + 1)(\eta + 1)} - \frac{1}{\eta^2}\right) = 0, \end{aligned}$$

where the first inequality follows from $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, $\sigma_a \geq \sigma_h \geq (1 + 1/\eta)\,\sigma_{h+1}$, and $\sigma_{h+1} \geq \sigma_b$; the second (strict) inequality follows from $1 > \gamma_a^*, \gamma_b^* > 0$, and $1 \geq \gamma_a^* + \delta, \gamma_b^* - \delta \geq 0$. Therefore $\sum_{i=1}^m \sigma_i^2/(\eta + \gamma_i^*) > \sum_{i=1}^m \sigma_i^2/(\eta + \zeta_i^*)$. This completes the proof. □

We summarize the main result of this section in the following theorem.

Theorem 3.6.1. *Let the problems **F**$_1$ and **F**$_2$ be defined in Eqs. (3.13) and (3.16), respectively, and let (U^*, M^*) be the optimal solution to **F**$_2$. Let $P_1 \in \mathbb{R}^{d \times d}$ be orthogonal consisting of the left singular vectors of U^*, and $\{\sigma_i\}_{i=1}^q$ be the corresponding non-zero singular values of U^* in non-increasing order. Let Θ^* consist of the first h column-vectors of P_1 corresponding to the largest h singular values. If $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, then the optimal solution to **F**$_1$ is given by (U^*, Θ^*).*

Proof. Since (U^*, M^*) is optimal to **F**$_2$, it follows from Theorem 3.5.1 that M^* can be expressed as $M^* = P_1 \Lambda P_1^T$, where $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d) \in \mathbb{R}^{d\times d}$ can be computed via Eq. (3.39). Given $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, we can verify that $\lambda_i = 1$ if $i \in \mathbb{N}_h$, and 0 otherwise (Lemma 3.6.1); therefore $M^* = \Theta^{*T}\Theta^*$, where $\Theta^* \in \mathbb{R}^{d\times h}$ corresponds to the first h column-vectors of P_1. Moreover, given a fixed $U \in \mathbb{R}^{d\times m}$ in **F**$_1$ and **F**$_2$ respectively, we have

$$\min_{\Theta^T\Theta \in \mathcal{M}_e, \Theta\Theta^T = I} G_1(U, \Theta) \geq \min_{M \in \mathcal{M}_c} G_2(U, M), \tag{3.44}$$

where $G_1(U, \Theta)$ and $G_2(U.M)$ are defined in Eqs. (3.12) and (3.17) respectively, and $\mathcal{M}_e$ and $\mathcal{M}_c$ are defined in Eqs. (3.14) and (3.15) respectively. The equality in Eq. (3.44) is attained when the optimal M to the right side of Eq. (3.44) is an extreme point of the set $\mathcal{M}_c$, i.e., belong to the set $\mathcal{M}_e$. For a given U^*, if $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$ is satisfied, Θ^* minimizes $G_1(U^*, \Theta)$ and the equality in Eq. (3.44) can be attained. Hence, (U^*, Θ^*) is the optimal solution to **F**$_1$. This completes the proof. □

3.7 Example: Multi-Task Learning with Hinge Loss Function

In this section, we present a concrete example to illustrate the APG algorithm and the CASO algorithm for solving rASO in Eq. (3.16). We employ the Hinge Loss function for the rASO formulation:

$$\begin{aligned} \min_{\{u_\ell\}, M} \quad & \sum_{\ell=1}^{m} \left(\sum_{i=1}^{n_\ell} L(u_\ell^T x_i^\ell, y_i^\ell) + c\, u_\ell^T (\eta I + M)^{-1} u_\ell \right) \\ \text{subject to} \quad & \text{tr}(M) = h,\ M \preceq I,\ M \in \mathbb{S}_+^d, \end{aligned} \tag{3.45}$$

where the loss function L is given by

$$L\left(u_\ell^T x_i^\ell, y_i^\ell\right) = \max\left(1 - y_i^\ell \left(u_\ell^T x_i^\ell + b^\ell\right), 0\right),$$

and the parameter c is given by $c = \alpha\, \eta\, (1 + \eta)$. Note that the optimization problem in Eq. (3.45) is non-smooth convex due to the non-smooth Hinge Loss function.

The APG Algorithm for Solving Eq. (3.45)

We employ the APG algorithm in Section 3.4 to solve Eq. (3.45). We present the computational procedures of solving the key subproblem in the APG algorithm and also present the detailed APG algorithm for solving Eq. (3.45).

Solving the Key Subproblem

To solve general non-smooth optimization problems, the APG algorithm solves a key subproblem in Eq. (3.33) in each of its iterations. For Eq. (3.45), the associated key subproblem can be expressed as

$$\begin{array}{rl} \min\limits_{U_Z, M_Z} & \left\|U_Z - \hat{U}_S\right\|_F^2 + \left\|M_Z - \hat{M}_S\right\|_F^2 + \hat{\gamma} \sum\limits_{\ell=1}^{m} \sum\limits_{i=1}^{n_\ell} \xi_{\ell i} \\ \text{subject to} & \xi_{\ell i} \geq 0,\ \xi_{\ell i} \geq 1 - y_i^\ell (u_\ell^T x_i^\ell + b_\ell), \\ & \text{tr}(M_Z) = h,\ M_Z \preceq I,\ M_Z \in \mathbb{S}_+^d, \end{array} \tag{3.46}$$

where $\hat{\gamma} = \frac{2}{\gamma}$. In Eq. (3.46), the optimization variables U_Z and M_Z are decoupled and the optimal solution can be obtained independently via solving two optimization problems as below.

Computation of U_Z The optimal U_Z to Eq. (3.46) can be computed via solving

$$\begin{array}{rl} \min\limits_{U_Z} & \left\|U_Z - \hat{U}_S\right\|_F^2 + \hat{\gamma} \sum\limits_{\ell=1}^{m} \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} & \xi_{\ell i} \geq 0, \xi_{\ell i} \geq 1 - y_i^\ell (u_\ell^T x_i^\ell + b_\ell). \end{array} \tag{3.47}$$

Let $U_Z = [u_1 \cdots u_m]$ and $\hat{U}_S = [\hat{u}_1 \cdots \hat{u}_m]$. Each of the vector u_ℓ can be obtained by solving a QP problem as

$$\begin{array}{rl} \min\limits_{u_\ell} & \|u_\ell - \hat{u}_\ell\|_F^2 + \hat{\gamma} \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} & \xi_{\ell i} \geq 0, \xi_{\ell i} \geq 1 - y_i^\ell (u_\ell^T x_i^\ell + b_\ell). \end{array}$$

Computation of M_Z The optimal M_Z to Eq. (3.46) can be computed via solving

$$\begin{array}{rl} \min\limits_{M_Z} & \left\|M_Z - \hat{M}_S\right\|_F^2 \\ \text{subject to} & \text{tr}(M_Z) = h,\ M_Z \preceq I,\ M_Z \in \mathbb{S}_+^d. \end{array} \tag{3.48}$$

The optimal M_Z can be obtained via two steps:

- **Step** 1 Compute the eigendecomposition of the symmetric $\hat{M}_S$ as $\hat{M}_S = P\hat{\Sigma}P^T$, where $P \in \mathbb{R}^{d\times d}$ is orthogonal, $\hat{\Sigma} = \text{diag}(\sigma_1, \cdots, \sigma_d) \in \mathbb{R}^{d\times d}$ is diagonal with the eigenvalues on its main diagonal.

- **Step** 2 Solve the optimization problem

$$\begin{array}{rl} \min\limits_{\{\sigma_i\}} & \sum\limits_{i=1}^{d} (\sigma_i - \hat{\sigma}_i)^2 \\ \text{subject to} & \sum\limits_{i=1}^{d} \sigma_i = h,\ 0 \leq \sigma_i \leq 1. \end{array}$$

  and denote its optimal solution by $\{\sigma_i^*\}$.

The optimal M_Z to Eq. (3.48) is given by $M_Z = P\Sigma^* P^T$, where $\Sigma^* = \text{diag}(\sigma_1^*, \cdots, \sigma_d^*) \in \mathbb{R}^{d\times d}$.

```
1: **Input:** $U_{Z0} \in \mathbb{R}^{d\times m}$, $M_{Z0} \in \mathbb{R}^{d\times d}$, $L_0 \in \mathbb{R}$, max-iter.
2: **Output:** $U_Z \in \mathbb{R}^{d\times m}$, $M_Z \in \mathbb{R}^{d\times d}$.
3: Set $U_{Z1} = U_{Z0}$, $M_{Z1} = M_{Z0}$, $t_{-1} = 0$, and $t_0 = 1$.
4: **for** $i = 1, 2, \cdots,$ max-iter **do**
5:     Compute $\alpha_i = (t_{i-2} - 1)/t_{i-1}$.
6:     Compute $U_{Si} = (1 + \alpha_i)U_{Zi} - \alpha_i U_{Zi-1}$.
7:     Compute $M_{Si} = (1 + \alpha_i)M_{Zi} - \alpha_i M_{Zi-1}$.
8:     **while** (**true**)
9:         Compute $\hat{U}_Z$ via Eq. (3.47).
10:        Compute $\hat{M}_Z$ via Eq. (3.48).
11:        **if** Eq. (3.49) holds **then** exit the loop
12:            **else** update $L_i = L_i \times 2$.
13:        **end-if**
14:    **end-while**
15:    Update $U_{Zi+1} = \hat{U}_Z$, $M_{Zi+1} = \hat{M}_Z$, $L_{i+1} = L_i$.
16:    **if** stopping criteria satisfied **then** exit the loop.
17:    Update $t_i = \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$.
18: **end-for**
19: Set $U_Z = U_{Zi+1}$ and $M_Z = M_{Zi+1}$.
```

Algorithm 5: Solve Eq. (3.45) via the APG Algorithm

The Main Algorithm

The pseudo-codes of the APG algorithm for solving Eq. (3.45) is presented in Algorithm 5. Algorithm 5 solves Eq. (3.46) in each of its iteration, where $\hat{\gamma}$ is set as $2/L_i$, where the value of L_i is determined via the line search scheme. Note that the line search condition in line 10 of Algorithm 5 can be expressed as

$$f(\hat{Z}) \leq f(S_i) + \langle \hat{Z} - S_i, \nabla_{S_i} f(S_i)\rangle + \frac{L_i}{2}\|\hat{Z} - S_i\|_F^2, \tag{3.49}$$

where the function $f(\cdot)$ is defined in Eq. (3.31), the composite variables $\hat{Z}$ and S_i can be expressed as

$$\hat{Z} = \begin{bmatrix} \hat{U}_Z \\ \hat{M}_Z \end{bmatrix}, \;\; S_i = \begin{bmatrix} U_{S_i} \\ M_{S_i} \end{bmatrix},$$

and the derivative $\nabla_{S_i} f(S_i)$ can be expressed as

$$\nabla_{S_i} f(S_i) = \begin{bmatrix} 2c\,(\eta I + M_{S_i})^{-1}\, U_{S_i} \\ -c\,(\eta I + M_{S_i})^{-1}\, U_{S_i} U_{S_i}^T\, (\eta I + M_{S_i})^{-1} \end{bmatrix}.$$

The CASO Algorithm for Solving Eq. (3.45)

We employ the CASO algorithm in Section 3.5 to solve Eq. (3.45). Similarly, we present the detailed pseudo-codes of CASO in Algorithm 6 and discuss its main computational procedures.

Optimization of U Given a fixed M, we can optimize the variable U as

$$\begin{array}{ll} \min\limits_{\{u_\ell, b_\ell\}} & \sum\limits_{\ell=1}^{m} \left(\sum\limits_{i=1}^{n_\ell} \xi_{\ell i} + c\, u_\ell^T \left(\eta I + M\right)^{-1} u_\ell \right) \\ \text{subject to} & \xi_{\ell i} \geq 0,\, \xi_{\ell i} \geq 1 - y_i^\ell (u_\ell^T x_i^\ell + b_\ell). \end{array}$$

Since all pairs of the variables $\{u_\ell, b_\ell\}$ ($\ell \in \mathbb{N}_m$) in the problem above are decoupled, we can optimize each of the pairs by solving a QP problem in the form of

$$\begin{array}{ll} \min\limits_{u,b} & \sum_{i=1}^{n} \xi_i + c\, u^T (\eta I + M)^{-1} u \\ \text{subject to} & \xi_i \geq 0,\; \xi_i \geq 1 - y_i (u^T x_i + b). \end{array} \tag{3.50}$$

The problem in Eq. (3.50) is a linearly constrained quadratic program; the hessian matrix of the objective function is generally dense and the number of optimization variables is dependent on both the dimensionality and the sample size. We convert Eq. (3.50) into its equivalent dual form, in which the number of optimization variables only depends on the sample size.

By augmenting the objective function of Eq. (3.50) with the constraints, we have the associated Lagrange function as

$$\begin{aligned} L = \sum_{i=1}^{n} \xi_i + c\, u^T (\eta I + M)^{-1} u - \sum_{i=1}^{n} \alpha_i \xi_i \\ - \sum_{i=1}^{n} \beta_i \left(\xi_i - 1 + y_i (u^T x_i + b) \right), \end{aligned}$$

where $\alpha_i, \beta_i \geq 0$ denote the dual variables. Taking derivatives with respective to the primal variables ξ_i, u, b and setting them equal to zero, we have

$$\frac{\partial L}{\partial \xi_i} = 1 - \alpha_i - \beta_i = 0, \tag{3.51}$$

$$\frac{\partial L}{\partial u} = 2c\, (\eta I + M)^{-1} u - \sum_{i=1}^{n} \beta_i y_i x_i = 0, \tag{3.52}$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^{n} \beta_i y_i = 0. \tag{3.53}$$

By substituting Eqs. (3.51), (3.52), (3.53) into Eq. (3.50), we have the dual optimization problem as

$$\begin{array}{ll} \min\limits_{u,b} & \beta^T e - \frac{1}{2} \beta^T \text{ diag}(y) \text{ Ker diag}(y)\ \beta \\ \text{subject to} & 0 \preceq \beta \preceq 1,\; \beta^T y = 0, \end{array} \tag{3.54}$$

where Ker $= \frac{1}{2c} X^T (\eta I + M) X \in \mathbb{R}^{n \times n}$.

Optimization of M Given a fixed U, we can optimize the variable M as

$$\begin{array}{ll} \min\limits_{M} & \text{tr} \left(U^T \left(\eta I + M\right)^{-1} U \right) \\ \text{subject to} & \text{tr}(M) = h,\; M \preceq I,\; M \in \mathbb{S}_+^d. \end{array} \tag{3.55}$$

The optimal M can be obtained via two steps:

- **Step** 1 Compute the SVD of U as $U = P_1 \Sigma P_2^T$, where $P_1 \in \mathbb{R}^{d \times d}$ and $P_2 \in \mathbb{R}^{m \times m}$ are orthogonal, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_q, 0, \cdots, 0) \in \mathbb{R}^{d \times m}$ has q non-zero singular values on its main diagonal.

- **Step** 2 Solve the optimization problem as

$$\begin{array}{ll} \min\limits_{\{\gamma_i\}_{i=1}^q} & \sum_{i=1}^q \frac{\sigma_i^2}{\eta + \gamma_i} \\ \text{subject to} & \sum_{i=1}^q \gamma_i = h,\ 0 \leq \gamma_i \leq 1. \end{array} \tag{3.56}$$

  and denote its optimal solution by $\{\gamma_i^*\}$.

The optimal M to Eq. (3.55) is given by $M = P_1 \Lambda^* P_2^T$, where $\Lambda^* = \text{diag}(\lambda_1^*, \cdots, \lambda_q^*, 0, \cdots, 0) \in \mathbb{R}^{d \times m}$.

```
Input: $\{(x_i^\ell, y_i^\ell)\}$, $i \in \mathbb{N}_{n_\ell}$, $\ell \in \mathbb{N}_m$, $h \in \mathbb{N}$.
Output: $U$, $V$, and $M$.
Parameter: $\alpha$ and $\beta$.
Initialize $M$ subject to the constraints in Eq. (3.45).
repeat
  Update $U$ via Eq. (3.54).
  Compute the SVD $U = P_1 \Sigma P_2^T$.
  Compute $\{\gamma_i^*\}_{i=1}^q$ via Eq. (3.56).
  Update $M$ as $M = P_1$diag$(\gamma_1^*, \cdots, \gamma_q^*) P_1^T$.
until convergence criterion is satisfied.
Construct $\Theta$ using the top $h$ eigenvectors of $M$.
Construct $V$ as $V = \Theta U$.
Return $U$, $V$ and $\Theta$.
```

Algorithm 6: Solve Eq. (3.45) via the CASO Algorithm

3.8 Experiments

In this section, we evaluate the proposed rASO (convex) formulation in Eq. (3.16) by a comparison with other representative MTL formulations using the yahoo web pages data sets and the *Drosophila* gene expression pattern images data sets; we also conduct numerical studies on the APG algorithm and the CASO algorithm for solving Eq. (3.16).

We use the Yahoo web pages data sets [29] in our first experiment. The Yahoo data sets consist of 11 top-level categories, where each top-level category corresponds to one data set. Each top-level category is further divided into a set of second-level sub-categories, where each second-level sub-category corresponds to a topic included in one data set (one top-level category). We preprocess the data sets by removing the topics with a small number (less than 100) of web pages. We extract the TF-IDF (Term Frequency-Inverse Document Frequency) features from the web pages and the obtained feature vectors are normalized to unit length. The statistics of the processed data sets are summarized in Table 4.1.

Table 3.1: Statistics of eleven Yahoo web page data sets.

Data Set	Sample Size	Dimension	Tasks
Arts	7441	17973	19
Business	9968	16621	17
Computers	12317	25259	23
Education	11817	20782	14
Entertainment	12619	27435	14
Health	9202	18430	14
Recreation	12797	25095	18
Reference	7929	26397	15
Science	6345	24002	22
Social	11914	32492	21
Science	14507	29189	21

Table 3.2: Performance comparison of the competing algorithms on six Yahoo data sets. In ASO and rASO, the shared feature dimensionality h is set as $\lfloor (m-1)/5 \rfloor \times 5$.

Data		Arts	Business	Computers	Education	Entertainment	Health
Macro F1	SVM	33.93 ± 1.07	44.43 ± 0.56	30.09 ± 1.10	39.00 ± 2.42	46.88 ± 0.47	56.14 ± 2.58
	ASO	37.93 ± 1.57	44.64 ± 0.40	28.33 ± 0.67	36.93 ± 1.98	47.46 ± 0.37	57.63 ± 0.74
	rASO	37.35 ± 0.60	45.79 ± 0.69	33.35 ± 0.84	41.28 ± 0.90	49.66 ± 0.97	61.16 ± 1.70
	cMTFL	37.06 ± 0.75	40.90 ± 1.66	32.50 ± 0.90	40.17 ± 0.55	50.94 ± 1.06	58.66 ± 2.22
Micro F1	SVM	43.99 ± 1.23	77.51 ± 0.51	55.36 ± 0.63	48.03 ± 1.56	55.69 ± 2.45	61.40 ± 4.76
	ASO	43.96 ± 0.03	78.08 ± 0.25	54.43 ± 0.40	46.97 ± 0.37	57.71 ± 0.33	65.90 ± 0.39
	rASO	47.69 ± 0.47	77.44 ± 0.94	54.54 ± 1.07	49.50 ± 0.57	57.90 ± 1.38	68.19 ± 1.01
	cMTFL	46.31 ± 0.32	69.00 ± 1.01	49.38 ± 4.22	48.56 ± 0.40	58.25 ± 0.76	66.83 ± 1.72

Table 3.3: Performance comparison of competing algorithms on five Yahoo data sets. Explanation can be found in Table 3.2.

Data Set		Recreation	Reference	Science	Social	Society
Macro F1	SVM	43.01 ± 1.44	39.37 ± 1.15	41.80 ± 1.45	35.87 ± 0.79	30.68 ± 0.94
	ASO	43.63 ± 1.29	37.46 ± 0.27	39.26 ± 0.82	35.29 ± 0.67	29.42 ± 0.30
	rASO	47.12 ± 0.73	42.11 ± 0.60	45.46 ± 0.50	39.30 ± 1.28	34.84 ± 1.05
	cMTFL	46.13 ± 0.58	43.25 ± 0.81	42.52 ± 0.59	38.94 ± 1.88	33.79 ± 1.43
Micro F1	SVM	49.15 ± 2.32	55.11 ± 3.16	49.27 ± 4.64	63.05 ± 2.45	40.07 ± 3.42
	ASO	50.68 ± 0.18	57.72 ± 0.51	49.05 ± 0.57	62.77 ± 3.59	46.13 ± 2.33
	rASO	53.34 ± 0.90	59.39 ± 0.39	53.32 ± 0.45	66.04 ± 0.62	49.27 ± 0.55
	cMTFL	52.52 ± 0.92	58.49 ± 0.51	50.60 ± 0.76	65.60 ± 0.63	46.46 ± 0.87

In our experiments, we treat one topic (the second-level sub-category) in the Yahoo web page data sets as one task, and apply the proposed rASO formulation to categorize the multi-topic web pages. We conduct all experiments using the rASO formulation with Hinge Loss, i.e., Eq. (3.45). The quadratic programs (QPs) in our experiments are solved via MOSEK[3].

Evaluation of rASO

We evaluate the performance of the rASO formulation and study the sensitivity of its parameters. In the following experiments, rASO is solved using CASO.

Performance Comparison We compare rASO with SVM (the independent SVM for multi-task learning), ASO (the alternating structure optimization) [16], and cMTFL (the convex multi-task feature learning) [53] for Yahoo web pages categorization tasks. Note that cMTFL is essentially equivalent to the approach of employing the trace norm as a regularization for multi-task learning [54–56]. We employ Macro F1 and Micro F1 [64] as the performance measures. The parameters in the competing algorithms (the penalty parameter C in SVM, the regularization parameters in ASO, rASO and cMTFL) are determined via 3-fold cross-validation. In ASO, rASO and cMTFL, we stop the iterative computational procedure if the relative change of the objective values in two successive iterations is smaller than 10^{-5}. We randomly choose 1500 data points from each Yahoo web pages data set as the training data, and the remaining are used as the test data.

We report the averaged Macro F1 and Micro F1 (over 5 random repetitions) and the associated standard deviation in Tables 3.2 and 3.3. We can observe that rASO is competitive in comparison with other competing algorithms on all of the 11 Yahoo data sets. We can also observe that rASO outperforms ASO (in this supervised setting) on 9 data sets (except on the Arts data and the Business data) in terms of both Macro F1 and Micro F1; this superiority may be due to the employment of the different regularizer in Eq. (3.7), the flexibility of balancing the two regularization components, and the guaranteed global optimal solution in rASO. The relatively low performance of SVM may be due to its ignorance of the relationship among the multiple learning tasks.

Sensitivity Study We study the effect of the parameter η on the performance of rASO, where $\eta = \beta/\alpha$ is defined in Eq. (3.10), α and β are used to trade off two regularization components in Eq. (3.7). We fix α at 1, vary β in the range of $[10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}]$, and record the obtained Macro/Micro F1, respectively. The Arts data is used for this experiment.

The experimental results are presented in Figure 3.1. We can observe that if the value of η is smaller, rASO achieves relatively low performance in terms of Macro F1 and Micro F1; if η is set

[3] http://www.mosek.com/

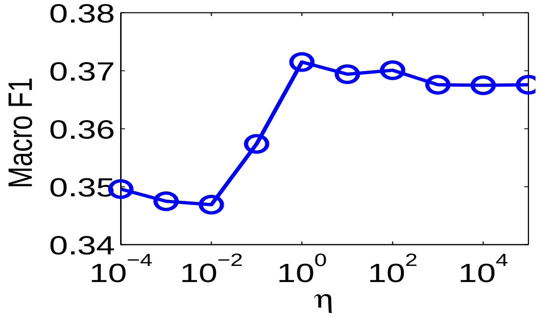


Figure 3.1: Sensitivity study of the parameter η in rASO: we study the relationship between the parameter η and the corresponding Micro $\mathrm{F}1$ and Micro $\mathrm{F}1$ obtained in rASO.

to some value close to 1, rASO can achieve the best performance. We observe a similar trend in other data sets. Since the value of η is equal to the ratio of β to α, our empirical observation (setting $\eta \approx 1$ leading to good performance) demonstrates that adding the second regularization component of Eq. (3.7) in appropriate amount (corresponding to the parameter β) can improve the performance.

Evaluation of APG and CASO

We evaluate the APG algorithm and the CASO algorithm in terms of the (global) convergence and the computation time (in seconds) by solving Eq. (3.45). For illustration, we set $\alpha = 1, \beta = 1, h = 1$ in Eq. (3.45) and use the Arts data for the following experiments; for other parameters settings, we have similar observations. Note that APG and CASO are terminated if the change of the objective values in two successive iterations is smaller than 10^{-5} or the iteration number is larger than 5000.

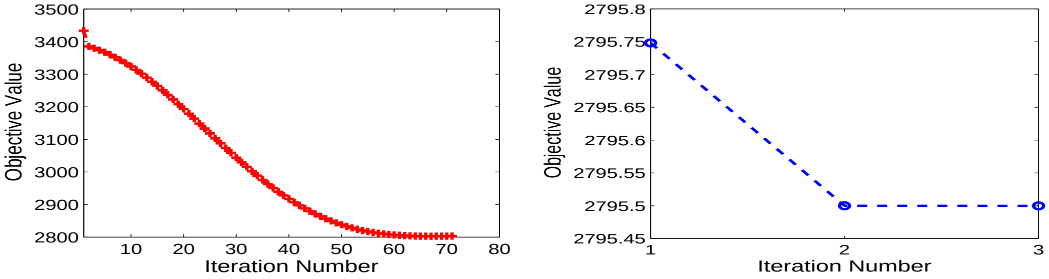


Figure 3.2: Convergence of APG (left plot) and CASO (right plot) for solving Eq. (3.45): we study the relationship between the objective value and the required iterations for attaining such a value.

Convergence Comparison We construct a subset for this experiment by randomly choosing 4000 samples from the Art data and selecting the first 3 topics (tasks); for illustration, we also reduce the feature dimensionality to 100 via PCA. We apply the APG algorithm and the CASO algorithm separately for solving Eq. (3.45) on the constructed subset and record the obtained objective value in each of the iterations.

The experimental results are presented in Figure 3.2. From Figure 3.2, we can observe that APG requires about 72 iterations for convergence and its convergence curve is consistent with the theoretical convergence analysis of the APG algorithm [45,57]. We can also observe that CASO converges very fast in practice; CASO converges within 3 iterations in this experiment (when the value of β is smaller than the value of α, CASO require a larger number of iterations for convergence). Note that in this experiment, the total computation time for APG and CASO are 121 seconds and 1653 seconds, respectively; although CASO requires much less number of iterations for convergence, the computation cost in each iteration of CASO is much higher than that of APG.

Table 3.4: Computation time (in seconds) comparison for APG and CASO. We fix the dimension at 100 and vary the sample size in the set $\{1000, 2000, 3000, 4000\}$.

Sample Size	Time_{APG}	$\text{Time}_{\text{CASO}}$	Time_{APG} : $\text{Time}_{\text{CASO}}$
1000	32.63	16.42	1.9867
2000	68.63	165.66	0.4143
3000	102.97	604.85	0.1702
4000	120.80	1653.85	0.0730

Computation Time Comparison We construct 4 subsets by randomly choosing $\{1000, 2000, 3000, 4000\}$ samples from the Arts data respectively, and use the first 3 topics (tasks) for this experiment. For simple illustration, we reduce the feature dimensionality in the subsets to 100 via PCA. We apply APG and CASO on the constructed subsets and record the respective computation time in seconds. The experimental results are presented in Table 3.4. We can observe that the computation time for APG and CASO increase with the increase of the sample size. We can also observe that by using a fixed feature dimensionality, when the sample size is relatively smaller, for example 1000, APG requires more computation time (for convergence) compared to CASO; when the sample size is relatively larger, for example $2000, 3000, 4000$, APG requires less computation time compared to CASO.

Table 3.5: Computation time (in seconds) comparison for APG and CASO. We fix the sample size at 2000 and vary the dimension in the set $\{100, 200, 300, 400\}$.

Dimension	Time_{APG}	$\text{Time}_{\text{CASO}}$	Time_{APG} : $\text{Time}_{\text{CASO}}$
100	72.23	162.11	0.4456
200	509.86	177.78	2.8679
300	1264.87	184.59	6.8523
400	1555.53	195.70	7.9485

One the other hand, we construct another 4 subsets by randomly choosing 2000 samples from the Arts data and then reduce the feature dimensionality to $\{100, 200, 300, 400\}$ via PCA; we use the first 3 topics (tasks) for our experiment. We apply APG and CASO on the constructed subsets and record the respective computation time. The experimental results are presented in Table 3.5. We can observe that the computation time for APG and CASO increase with the increase of the feature dimensionality. We can also observe that by using a fixed sample size, when the feature dimensionality is relatively smaller, for example 100, APG requires less computation time compared to CASO; when the feature dimensionality is relatively larger, for example $200, 300, 400$, APG requires more computation time compared to CASO.

Table 3.6: Comparison of the optimal objective values for Eq. (3.6) and Eq. (3.16) with different η. We fix $\alpha = 1$ and vary the value of β in the range $[10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}]$.

$\eta = \beta/\alpha$	1000	100	10	1	0.1	0.01	0.001
$1 + 1/\eta$	1.001	1.01	1.1	2	11	101	1001
σ_h/σ_{h+1}	1.23	1.25	1.34	1.75	3.07	13.79	89.49
OBJ$_{\mathbf{F}_0}$	52.78	52.65	51.37	40.73	22.15	5.95	0.69
OBJ$_{\mathbf{F}_2}$	52.78	52.65	51.37	40.71	20.73	4.11	0.41

Empirical Comparison of $\mathbf{F}_0$ and $\mathbf{F}_2$

We compare $\mathbf{F}_0$ in Eq. (3.6) and $\mathbf{F}_2$ in Eq. (3.16) in terms of the obtained optimal objective values. Recall that in Eq. (3.10), we have $\eta = \beta/\alpha$. We vary the value of η by fixing $\alpha = 1$ and varying β in the range $[10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}]$, and compute the optimal objective values of $\mathbf{F}_0$ and $\mathbf{F}_2$, respectively. We randomly sample 500 data points from Arts data for this experiment. Both $\mathbf{F}_0$ and $\mathbf{F}_2$ are solved via the alternating optimization algorithm, i.e., optimizing one variable with the other variables fixed.

The experimental results are presented in Table 3.6. We can observe that OBJ$_{\mathbf{F}_2}$ is always no larger than OBJ$_{\mathbf{F}_0}$; this is because $\mathbf{F}_2$ is a relaxed version of $\mathbf{F}_2$ and has a larger domain set compared to $\mathbf{F}_0$. We also observe that if $\sigma_h/\sigma_{h+1} > 1 + 1/\eta$ (the second row of Table 3.6), OBJ$_{F_0}$ is equal to OBJ$_{\mathbf{F}_2}$ and the optimal solution to $\mathbf{F}_0$ can be recovered from $\mathbf{F}_2$; in general the condition $\sigma_h/\sigma_{h+1} > 1 + 1/\eta$ is satisfied when the value of β is relatively larger than the value of α. All of the observations are consistent with our theoretical analysis in Theorem 3.6.1.

Automated Annotation of the Gene Expression Pattern Images

In our second experiment, we apply the proposed rASO formulation for the automated annotation of the *Drosophila* gene expression pattern images from the FlyExpress[4] database. We use SVM (the independent SVM for multi-task learning), ASO (the alternating structure optimization) [16],

[4] `http://www.flyexpress.net/`

Table 3.7: Performance Comparison on competing algorithms for the gene expression pattern images annotation (10 CV terms) in terms of Macro F1 (top section) and Micro F1 (bottom section). In the second row, n, d, and m denotes sample size, dimension, and task numbers, respectively. In ASO and rASO, the shared feature dimensionality h is set as $\lfloor (m-1)/5 \rfloor \times 5$.

Stage Range		$4 \sim 6$	$7 \sim 8$	$9 \sim 10$	$11 \sim 12$	$13 \sim 16$
(n, d, m)		$(925, 2000, 10)$	$(797, 2000, 10)$	$(919, 2000, 10)$	$(1622, 2000, 10)$	$(2228, 2000, 20)$
Macro F1	SVM	40.88 ± 0.49	46.73 ± 0.51	50.28 ± 0.65	59.82 ± 0.83	59.62 ± 0.94
	ASO	43.29 ± 0.46	48.82 ± 0.62	51.55 ± 0.90	62.15 ± 0.16	60.11 ± 0.32
	rASO	44.54 ± 0.79	50.59 ± 0.23	54.16 ± 0.75	63.43 ± 0.79	60.90 ± 0.77
	cMTFL	42.21 ± 0.69	48.17 ± 0.65	52.22 ± 0.35	62.17 ± 1.03	60.12 ± 0.27
Micro F1	SVM	42.05 ± 0.61	60.09 ± 0.78	60.57 ± 0.75	67.08 ± 0.99	65.95 ± 0.80
	ASO	45.89 ± 0.33	61.15 ± 0.57	63.01 ± 0.52	67.91 ± 0.51	66.53 ± 0.25
	rASO	47.34 ± 0.18	62.77 ± 0.61	64.37 ± 0.19	70.61 ± 1.21	67.13 ± 1.01
	cMTFL	46.07 ± 0.92	60.35 ± 0.31	63.22 ± 0.67	68.43 ± 0.25	67.35 ± 0.59

Table 3.8: Performance Comparison on competing algorithms for the gene expression pattern images annotation (20 CV terms).

Stage Range		$4 \sim 6$	$7 \sim 8$	$9 \sim 10$	$11 \sim 12$	$13 \sim 16$
(n, d, m)		$(1023, 2000, 20)$	$(827, 2000, 20)$	$(1015, 2000, 20)$	$(1940, 2000, 20)$	$(2476, 2000, 20)$
Macro F1	SVM	29.47 ± 0.46	28.85 ± 0.62	30.03 ± 1.68	41.63 ± 0.58	40.80 ± 0.66
	ASO	30.33 ± 0.91	30.01 ± 0.67	32.22 ± 0.79	41.77 ± 1.43	40.98 ± 0.76
	rASO	31.01 ± 0.75	32.27 ± 0.91	35.01 ± 1.12	45.12 ± 0.21	43.81 ± 0.46
	cMTFL	30.66 ± 0.24	30.84 ± 0.39	34.13 ± 0.87	44.73 ± 0.49	43.13 ± 0.65
Micro F1	SVM	39.24 ± 0.82	55.40 ± 0.15	55.75 ± 0.70	58.33 ± 0.53	53.61 ± 0.36
	ASO	41.11 ± 0.32	57.72 ± 0.51	53.29 ± 0.21	61.77 ± 1.09	53.45 ± 0.92
	rASO	41.21 ± 1.24	59.34 ± 0.39	59.81 ± 0.33	63.25 ± 0.71	54.93 ± 0.78
	cMTFL	40.79 ± 0.31	58.39 ± 1.11	58.12 ± 0.84	61.22 ± 0.21	54.60 ± 0.62

and cMTL (the convex Multi-task feature learning) [53] as baseline algorithms. Note that in the following experiments, Hinge Loss is employed as the loss function for all of the competing algorithms.Similarly, we preprocess the *Drosophila* gene expression pattern images (of the standard size 128×320) from the FlyExpress database following the procedures in [27]. The *Drosophila* images are from 16 specific stages, which are then grouped into 6 stage ranges ($1 \sim 3$, $4 \sim 6$, $7 \sim 8$, $9 \sim 10$, $11 \sim 12$, $13 \sim 16$). We manually annotate the image groups (based on the genes and the developmental stages) using the structured CV terms. Each image group is then represented as a feature vector based on the bag-of-words and the soft-assignment sparse coding schemes. Note that the SIFT (scale-invariant feature transform) features [28] are extracted from the images with the patch size set at 16×16 and the number of visual words in sparse coding set at 2000. The first stage range only contains 2 CV terms and we do not report the performance for this stage range. For other stage ranges, we consider the top 10 and 20 CV terms that appear most frequently in the image groups and treat the annotation of each CV term as one task. We generate 10 subsets for this experiment, and randomly partition each subset into training and test sets using the ratio

$1:9$. Note that the parameters in the competing algorithms are tuned via 5-fold cross-validation as in Section 4.7.

We report the averaged Macro F1 and Micro F1 over 10 random repetitions in Table 3.7 (for 10 CV terms) and Table 3.8 (for 20 CV terms), respectively. We can observe that rASO performs the best or competitively compared to other representative algorithms on all subsets. This observation demonstrates the effectiveness of the proposed rASO formulation for the images annotation tasks in multi-task learning setting; it also implies the effectiveness of the proposed regularizer in Eq. (3.7) for capturing the relationship of different CV terms of the gene expression pattern images. We can also observe that rASO outperforms ASO in all subsets, which further provides strong support for our rationale of improving the ASO formulation using the regularizer in Eq. (3.7).

3.9 Summary

We present a multi-task learning formulation (iASO) for learning a shared feature representation from multiple related tasks. Since iASO is non-convex, we convert it into a relaxed convex formulation (rASO). In addition, we present a theoretical condition, under which rASO can find a globally optimal solution to iASO. We propose two algorithms including the APG algorithm and the CASO algorithm to find the globally optimal solution to rASO; we also develop efficient algorithms for solving the key subproblems involved in APG and CASO. Our analysis shows that the computational cost in APG mainly depends on the feature dimensionality, while the computational cost in CASO mainly depends on the sample size. We have conducted experiments on the yahoo web pages data sets and the *Drosophila* gene expression pattern images data sets. The experimental results demonstrate the effectiveness and efficiency of the proposed algorithms and confirm our theoretical analysis.

Chapter 4

Learning Incoherent Sparse and Low-rank Patterns from Multiple Tasks

4.1 Introduction

In many real-world applications, the underlying predictive classifiers may lie in a hypothesis space of some low-rank structure [16], in which the multiple learning tasks can be coupled using a set of shared factors, i.e., the basis of a low-rank subspace [65]. For example, in natural scene categorization problems, images of different labels may share similar background of a low-rank structure; in collaborative filtering or recommendation system, only a few factors contribute to an individual's tastes. On the other hand, multiple learning tasks may exhibit sufficient differences and meanwhile the discriminative features for each task can be sparse. Thus learning an independent predictive classifier for each task and identifying the task-relevant discriminative features simultaneously may lead to improved performance and easily interpretable models.

We consider the problem of learning incoherent sparse and low-rank patterns from multiple related tasks. We propose a linear multi-task learning formulation, in which the model parameter can be decomposed as a sparse component and a low-rank component. Specifically, we employ a cardinality regularization term to enforce the sparsity in the model parameter, identifying the essential discriminative feature for effective classification; meanwhile, we use a rank constraint to encourage the low-rank structure, capturing the underlying relationship among the tasks for improved generalization performance. The proposed multi-task learning formulation is non-convex and leads to an NP-hard optimization problem. We convert this formulation into its tightest convex surrogate, which can be routinely solved via semi-definite programming. It is, however, not scalable to large scale data sets in practice. We propose to employ the general projected gradient scheme to solve the convex surrogate; however, in the optimization formulation, the objective function is non-differentiable and the feasible domain is non-trivial. We present the procedures for computing the projected gradient and ensuring the global convergence of the projected gradient scheme. The computation of projected gradient involves a constrained optimization problem; we show that the optimal solution to such a problem can be obtained via solving an unconstrained optimization subproblem and an Euclidean projection subproblem separately. We also present two algorithms based on the projected gradient scheme and analyze their rates of convergence in details. In addition, we present an example of the proposed multi-task learning formulation using the least squares loss and illustrate the use of the presented projected gradient based algorithms in this case. We conduct extensive experiments on a collection of real-world data sets. Our results demonstrate the effectiveness of the proposed multi-task learning formulation and also demonstrate the efficiency of the projected gradient algorithms.

The remainder of this chapter is organized as follows: in Section 5.2 we propose the linear multi-task learning formulation; in Section 4.3 we present the general projected gradient scheme for solving the proposed multi-task learning formulation; in Section 4.4 we present efficient computational algorithms for solving the optimization problems involved in the iterative procedure of the projected gradient scheme; in Section 6.3 we present two algorithms based on the projected gradient scheme and analyze their rates of convergence in details; in Section 4.6 we present a concrete example on the use of the projected gradient based algorithms for the proposed multi-task learning formulation using the least squares loss; we report the experimental results in Section 4.7 and this chapter concludes in Section 4.8.

4.2 Multi-Task Learning Framework

Assume that we are given m supervised (binary) learning tasks, where each of the learning tasks is associated with a predictor f_ℓ and a set of training data as $\{(x_i^\ell, y_i^\ell)\}_{i=1}^{n_\ell} \subset \mathbb{R}^d \times \{-1, +1\}$ $(\ell = 1, \cdots, m)$. We focus on linear predictors as $f_\ell(x^\ell) = z_\ell^T x^\ell$, where $z_\ell \in \mathbb{R}^d$ is the weight vector for the ℓth learning task.

We assume that the m tasks are related using an incoherent rank-sparsity structure, that is, the transformation matrix can be decomposed as a sparse component and a low-rank component. Denote the transformation matrix by $Z = [z_1, \cdots, z_m] \in \mathbb{R}^{d\times m}$; Z is the summation of a sparse matrix $P = [p_1, \cdots, p_m] \in \mathbb{R}^{d\times m}$ and a low-rank matrix $Q = [q_1, \cdots, q_m] \in \mathbb{R}^{d\times m}$ given by

$$Z = P + Q, \tag{4.1}$$

as illustrated in Figure 4.1. The ℓ^0-norm (cardinality) [38], i.e., the number of non-zero entries, is commonly used to control the sparsity structure in the matrix; similarity, matrix rank [49] is used to encourage the low-rank structure. We propose a multi-task learning formulation with a cardinality regularization and a rank constraint given by

$$\begin{aligned} \min_{Z,P,Q\in\mathbb{R}^{d\times m}} \quad & \sum_{\ell=1}^{m}\sum_{i=1}^{n_\ell} \mathcal{L}\left(z_\ell^T x_i^\ell, y_i^\ell\right) + \gamma\|P\|_0 \\ \text{subject to} \quad & Z = P + Q, \ \text{rank}(Q) \leq \tau, \end{aligned} \tag{4.2}$$

where $\mathcal{L}(\cdot)$ denotes a smooth convex loss function, γ provides a trade-off between the sparse regularization term and the general loss component, and τ explicitly specifies the upper bound of the

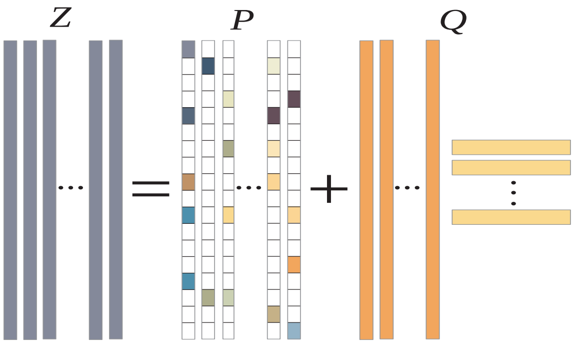


Figure 4.1: Illustration of the transformation matrix Z in Eq. (4.1), where P denotes the sparse component with the zero-value entries represented by white blocks, and Q denotes the low-rank component.

matrix rank. Both γ and τ are non-negative and determined via cross-validation in our empirical studies.

The optimization problem in Eq. (4.2) is non-convex due to the non-convexity of the components $\|P\|_0$ and rank(Q); in general solving such an optimization problem is NP-hard and no efficient solution is known. We consider a computationally tractable alternative by employing recently well-studied convex relaxation techniques [38].

Define the function $f : \mathbb{C} \to \mathbb{R}$, where $\mathbb{C} \subseteq \mathbb{R}^{d \times m}$. The convex envelope [38] of f on $\mathbb{C}$ is defined as the largest convex function g such that $g(\hat{Z}) \leq f(\hat{Z})$ for all $\hat{Z} \in \mathbb{C}$. The ℓ^1-norm has been known as the convex envelope of the ℓ^0-norm as [38]:

$$\|P\|_1 \leq \|P\|_0, \ \forall P \in \mathbb{C} = \{P \,|\, \|P\|_\infty \leq 1\}. \tag{4.3}$$

Similarly, trace norm (nuclear norm) has been shown as the convex envelop of the rank function as [66]:

$$\|Q\|_* \leq \text{rank}(Q), \ \forall Q \in \mathbb{C} = \{Q \,|\, \|Q\|_2 \leq 1\}. \tag{4.4}$$

Note that both the ℓ^1-norm and the trace-norm functions are convex but non-smooth, and they have been shown to be effective surrogates of the ℓ^0-norm and the matrix rank functions, respectively.

Based on the heuristic approximations in Eq. (4.3) and Eq. (4.4), we can replace the ℓ^0-norm with the ℓ^1-norm, and replace the rank function with the trace norm function in Eq. (4.2), respectively. Therefore, we can reformulate the multi-task learning formulation as:

$$\begin{aligned} \min_{Z,P,Q \in \mathbb{R}^{d \times m}} \quad & \sum_{\ell=1}^{m} \sum_{i=1}^{n_\ell} \mathcal{L}\left(z_\ell^T x_i^\ell, y_i^\ell\right) + \gamma \|P\|_1 \\ \text{subject to} \quad & Z = P + Q, \ \|Q\|_* \leq \tau. \end{aligned} \tag{4.5}$$

The optimization problem in Eq. (4.5) is the tightest convex relaxation of Eq. (4.2). Such a problem can be reformulated as a semi-definite program (SDP) [67], and solved using many off-the-shelf optimization solvers such as SeDuMi [41]; however, SDP is computationally expensive and can only handle several hundreds of optimization variables.

Related Work The formulation in Eq. (4.5) resembles the Alternating Structure Optimization algorithm (ASO) for multi-task learning proposed in [16]. However, they differ in several key aspects: (1) In ASO, the tasks are coupled using a shared low-dimensional structure induced by an orthonormal constraint, and the formulation in ASO is non-convex and its convex counterpart cannot be easily obtained. Our formulation encourages the low-rank structure via a trace norm constraint and the resulting formulation is convex. (2) In ASO, in addition to a low-dimensional feature map shared by all tasks, the classifier for each task computes an independent high-dimensional feature map specific to each individual task, which is in general dense and does not lead to interpretable features. In our formulation, the classifier for each task constructs a sparse high-dimensional feature map for discriminative feature identification. (3) The alternating algorithm in ASO can only find a local solution with no known convergence rate. The proposed algorithm for solving the formulation in Eq. (4.5) finds a globally optimal solution and achieves the optimal convergence rate among all first-order methods. Note that recent works in [68–70] consider the problem of decomposing a given matrix into its underlying sparse component and low-rank component in a different setting: they study the theoretical condition under which such two components can be exactly recovered via convex optimization, i.e., the condition of guaranteeing to recover the sparse and low-rank components by minimizing a weighted combination of the trace norm and the ℓ^1-norm.

4.3 Projected Gradient Scheme

In this section, we propose to apply the general projected gradient scheme [38] to solve the constrained optimization problem in Eq. (4.5). Note that the projected gradient scheme belongs to the category of the first-order methods and has demonstrated good scalability in many optimization problems [38, 57].

The objective function in Eq. (4.5) is non-smooth and the feasible domain is non-trivial. For simplicity, we denote Eq. (4.5) as

$$\begin{aligned} \min_{T} \quad & f(T) + g(T) \\ \text{subject to} \quad & T \in \mathcal{M}, \end{aligned} \tag{4.6}$$

where the functions $f(T)$ and $g(T)$ are defined respectively as

$$f(T) = \sum_{\ell=1}^{m} \sum_{i=1}^{n_\ell} \mathcal{L}\left((p_\ell + q_\ell)^T x_i^\ell, y_i^\ell\right), \;\; g(T) = \gamma \|P\|_1,$$

and the set $\mathcal{M}$ is defined as

$$\mathcal{M} = \left\{ T \,\middle|\, T = \begin{pmatrix} P \\ Q \end{pmatrix}, \; P \in \mathbb{R}^{d \times m}, \; \|Q\|_* \leq \tau, \; Q \in \mathbb{R}^{d \times m} \right\}.$$

Note that $f(T)$ is a smooth convex function with a Lipschitz constant L_f [63] as:

$$\|\nabla f(T_x) - \nabla f(T_y)\|_F \leq L_f \|T_x - T_y\|_F, \, \forall T_x, T_y \in \mathcal{M}, \tag{4.7}$$

$g(T)$ is a non-smooth convex function, and $\mathcal{M}$ is a compact and convex set [63]. It is known that the smallest Lipschitz constant $\hat{L}_f$ in Eq. (4.7), i.e, $\hat{L}_f = \min L_f$, is called the best Lipschitz constant for the function $f(T)$; moreover, for any $L \geq \hat{L}_f$, the following inequality holds [45]:

$$f(T_x) \leq f(T_y) + \langle T_x - T_y, \nabla f(T_y) \rangle + \frac{L}{2} \|T_x - T_y\|^2, \tag{4.8}$$

where $T_x, T_y \in \mathcal{M}$.

The projected gradient scheme computes the global minimizer of Eq. (4.6) via an iterative refining procedure. That is, given T_k as the intermediate solution of the kth iteration, we refine T_k as

$$T_{k+1} = T_k - t_k \mathcal{P}_k, \; \forall k, \tag{4.9}$$

where $\mathcal{P}_k$ and t_k denote the appropriate projected gradient direction and the step size, respectively. The appropriate choice of $\mathcal{P}_k$ and t_k is key to the global convergence of the projected gradient scheme. The computation of Eq. (4.9) depends on $\mathcal{P}_k$ and t_k; in the following subsections, we will present a procedure for estimating appropriate $\mathcal{P}_k$ and t_k, and defer the discussion of detailed projected gradient based algorithms to Section 6.3. Note that since the determination of $\mathcal{P}_k$ is associated with T_k and t_k, we denote $\mathcal{P}_k$ by $\mathcal{P}_{1/t_k}(T_k)$, and the reason will become clear from the following discussion.

Projected Gradient Computation

For any $L > 0$, we consider the construction associated with the smooth component $f(T)$ of the objective function in Eq. (4.6) as

$$f_L(S, T) = f(S) + \langle T - S, \nabla f(S) \rangle + \frac{L}{2} \|T - S\|_F^2,$$

where $S, T \in \mathbb{R}^{d \times m}$. It can be verified that $f_L(S,T)$ is strongly convex with respect to the variable T. Moreover, we denote

$$G_L(S,T) = f_L(S,T) + g(T), \tag{4.10}$$

where $g(T)$ is the non-smooth component of the objective function in Eq. (4.6). From the convexity of $g(T)$, $G_L(S,T)$ is strongly convex with respect to T. Since

$$G_L(S,T) \quad = \quad f(S) - \frac{1}{2L}\|\nabla f(S)\|_F^2 + \frac{L}{2}\left\|T - \left(S - \frac{1}{L}\nabla f(S)\right)\right\|_F^2 + g(T),$$

the global minimizer of $G_L(S,T)$ with respect to T can be computed as

$$\begin{aligned} T_{L,S} &= \arg\min_{T \in \mathcal{M}} G_L(S,T) \\ &= \arg\min_{T \in \mathcal{M}} \left(\frac{L}{2}\left\|T - \left(S - \frac{1}{L}\nabla f(S)\right)\right\|_F^2 + g(T)\right). \end{aligned} \tag{4.11}$$

Therefore we can obtain the projected gradient of f at S via

$$\mathcal{P}_L(S) = L(S - T_{L,S}). \tag{4.12}$$

It is obvious that $1/L$ can be seen as the step size associated with the projected gradient $\mathcal{P}_L(S)$ by rewritting Eq. (4.12) as

$$T_{L,S} = S - \frac{1}{L}\mathcal{P}_L(S). \tag{4.13}$$

Note that if the inequality $f(T_{L,S}) \leq f_L(S, T_{L,S})$ is satisfied, $\mathcal{P}_L(S)$ is called the L-projected gradient [57] of f at S.

Step Size Estimation

From Eq. (4.12), the step size associated with $\mathcal{P}_L(S)$ is given by $1/L$. Denote the objective function in Eq. (4.6) as

$$F(T) = f(T) + g(T). \tag{4.14}$$

Theoretically, any step size $1/L$ of the value L larger than the best Lipschitz constant $\hat{L}_f$ guarantees the global convergence in the projected gradient based algorithms [57]. It follows from Eq. (4.8) that

$$F(T_{L,S}) \leq G_L(S, T_{L,S}), \ \forall L \geq L_f. \tag{4.15}$$

In practice we can estimate an appropriate L (hence the appropriate step size $1/L$) by ensuring the inequality in Eq. (4.15). By applying an appropriate step size and the associated projected gradient in Eq. (4.9), we can verify an important inequality [57, 71], as summarized in the following lemma.

Lemma 4.3.1. *Let L_f be the Lipschitz continuous gradient associated with the function $f(T)$ as defined in Eq. (4.7). Let $S \in \mathbb{R}^{d\times m}$, and $T_{L,S}$ be the minimizer to $G_L(S,T)$ as defined in Eq. (4.11). Then if $L \geq L_f$, the following inequality holds*

$$F(T) - F(T_{L,S}) \geq \langle T - S, \mathcal{P}_L(S)\rangle + \frac{1}{2L}\|\mathcal{P}_L(S)\|_F^2 \tag{4.16}$$

for any $T \in \mathcal{M}$.

Proof. Following from the convexity of $f(\cdot)$ and $g(\cdot)$, we have

$$f(T) \quad \geq \quad f(S) + \langle T - S, \nabla f(S)\rangle \tag{4.17}$$

$$g(T) \quad \geq \quad g(T_{L,S)} + \langle T - T_{L,S}, \partial g(T_{L,S})\rangle\,, \tag{4.18}$$

where $\partial g(T_{L,S})$ denotes the subgradient [45] of $g(\cdot)$ at $T_{L,S}$. It is well known that $\hat{T}$ minimizes $G_L(S,T)$ (with respect to the variable T) if and only if **0** is a subgradient of $G_L(S,T)$ at $\hat{T}$, that is,

$$\mathbf{0} \in L\,(T_{L,S} - S) + \nabla f(S) + \partial g(T_{L,S}). \tag{4.19}$$

From Eqs. (4.10), (6.2), (4.17) and (4.18), we have

$$\begin{aligned}
F(T) - G_L(S, T_{L,S}) &= (f(T) + g(T)) - (f_L(S, T_{L,S}) + g(T_{L,S})) \\
&\geq \langle T - T_{L,S}, \nabla f(S) + \partial g(T_{L,S})\rangle - \frac{L}{2}\|S - T_{L,S}\|_F^2 \\
&= -L\,\langle T - T_{L,S}, T_{L,S} - S\rangle - \frac{L}{2}\|S - T_{L,S}\|_F^2 \\
&= \langle T - S, \mathcal{P}_L(S)\rangle + \frac{1}{2L}\|\mathcal{P}_L(S)\|_F^2\,,
\end{aligned}$$

where the second equality follows from Eq. (4.19), and the third equality follows from Eq. (4.12). This completes the proof of this lemma. □

By replacing S with T in Eq. (4.16), we have

$$F(T) - F(T_{L,T}) \geq \frac{1}{2L}\|\mathcal{P}_L(T)\|_F^2. \tag{4.20}$$

Note that the inequality in Eq. (4.16) characterizes the relationship of the objective values in Eq. (4.6) using T and its refined version via the procedure in Eq. (4.9).

4.4 Efficient Computation

The projected gradient scheme requires to solve Eq. (4.11) at each iterative step. In Eq. (4.11), the objective function is non-smooth and the feasible domain set is non-trivial; we show that its optimal solution can be obtained by solving an unconstrained optimization problem and an Euclidean projection problem separately.

Denote T and S in Eq. (4.11) respectively as

$$T = \begin{pmatrix} T_P \\ T_Q \end{pmatrix}, \quad S = \begin{pmatrix} S_P \\ S_Q \end{pmatrix}.$$

Therefore the optimization problem in Eq. (4.11) can be expressed as

$$\begin{array}{ll} \min\limits_{T_P, T_Q} & \dfrac{L}{2} \left\| \begin{pmatrix} T_P \\ T_Q \end{pmatrix} - \begin{pmatrix} \hat{S}_P \\ \hat{S}_Q \end{pmatrix} \right\|_F^2 + \gamma \|T_P\|_1 \\ \text{subject to} & \|T_Q\|_* \leq \tau, \end{array} \tag{4.21}$$

where $\hat{S}_P$ and $\hat{S}_Q$ can be computed respectively as

$$\hat{S}_P = S_P - \frac{1}{L} \nabla_P f(S), \ \hat{S}_Q = S_Q - \frac{1}{L} \nabla_Q f(S).$$

Note that $\nabla_P f(S)$ and $\nabla_Q f(S)$ denote the derivative of the smooth component $f(S)$ with respect to the variables P and Q, respectively. We can further rewrite Eq. (4.21) as

$$\begin{array}{ll} \min\limits_{T_P, T_Q} & \beta \|T_P - \hat{S}_P\|_F^2 + \beta \|T_Q - \hat{S}_Q\|_F^2 + \gamma \|T_P\|_1 \\ \text{subject to} & \|T_Q\|_* \leq \tau, \end{array} \tag{4.22}$$

where $\beta = L/2$. Since T_P and T_Q are decoupled in Eq. (4.22), they can be optimized separately as presented in the following subsections.

Computation of T_P

The optimal T_P to Eq. (4.22) can be obtained by solving the following optimization problem:

$$\min_{T_P} \ \beta \|T_P - \hat{S}_P\|_F^2 + \gamma \|T_P\|_1.$$

It is obvious that each entry of the optimal matrix T_P can be obtained by solving an optimization problem as

$$\min_{\hat{t} \in \mathbb{R}} \ \beta \|\hat{t} - \hat{s}\|^2 + \gamma |\hat{t}|. \tag{4.23}$$

Note that $\hat{s}$ denotes an entry in $\hat{S}_P$, corresponding to $\hat{t}$ in T_P from the same location. It is known [72] that the optimal $\hat{t}$ to Eq. (4.23) admits an analytical solution; for completeness, we present its proof in Lemma 4.4.1.

Lemma 4.4.1. *The minimizer of Eq. (4.23) can be expressed as*

$$\hat{t}^* = \begin{cases} \hat{s} - \frac{\gamma}{2\beta} & \hat{s} > \frac{\gamma}{2\beta} \\ 0 & -\frac{\gamma}{2\beta} \leq \hat{s} \leq \frac{\gamma}{2\beta} \\ \hat{s} + \frac{\gamma}{2\beta} & \hat{s} < -\frac{\gamma}{2\beta} \end{cases} . \tag{4.24}$$

Proof. Denote by $h(\hat{t})$ the objective function in Eq. (4.23), and by $\hat{t}^*$ the minimizer of $h(\hat{t})$. The subdifferential of $h(\hat{t})$ can be expressed as

$$\partial h(\hat{t}) = 2\beta(\hat{t} - \hat{s}) + \gamma \mathsf{sgn}(\hat{t}),$$

where the function sgn$(\cdot)$ is given by

$$\mathsf{sgn}(\hat{t}) = \begin{cases} \{1\} & \hat{t} > 0 \\ [-1, 1] & \hat{t} = 0 \\ \{-1\} & \hat{t} < 0 \end{cases} .$$

It is known that $\hat{t}^*$ minimizes $h(\hat{t})$ if and only if 0 is a subgradient of $h(\hat{t})$ at the point $\hat{t}^*$, that is,

$$0 \in 2\beta(\hat{t}^* - \hat{s}) + \gamma \mathsf{sgn}(\hat{t}^*).$$

Since the equation above is satisfied with $\hat{t}^*$ defined in Eq. (4.24), we complete the proof of this lemma. □

Computation of T_Q

The optimal T_Q to Eq. (4.22) can be obtained by solving the optimization problem:

$$\begin{aligned} \min_{T_Q} \quad & \frac{1}{2}\|T_Q - \hat{S}_Q\|_F^2 \\ \text{subject to} \quad & \|T_Q\|_* \leq \tau, \end{aligned} \tag{4.25}$$

where the constant $1/2$ is added into the objective function for convenient presentation. In the following theorem, we show that the optimal T_Q to Eq. (4.25) can be obtained via solving a simple convex optimization problem.

Theorem 4.4.1. *Let $\hat{S}_Q = U\Sigma_S V^T \in \mathbb{R}^{d\times m}$ be the SVD of $\hat{S}_Q$, where $q =$ rank$(\hat{S}_Q)$, $U \in \mathbb{R}^{d\times q}$, $V \in \mathbb{R}^{m\times q}$, and $\Sigma_S =$ diag$(\varsigma_1, \cdots, \varsigma_q) \in \mathbb{R}^{q\times q}$. Let $\{\sigma_i\}_{i=1}^q$ be the minimizers of the problem:*

$$\begin{aligned} \min_{\{\sigma_i\}_{i=1}^q} \quad & \textstyle\sum_{i=1}^q (\sigma_i - \varsigma_i)^2 \\ \textit{subject to} \quad & \textstyle\sum_{i=1}^q \sigma_i \leq \tau, \ \sigma_i \geq 0. \end{aligned} \tag{4.26}$$

Denote $\Sigma =$ diag$(\sigma_1, \cdots, \sigma_q) \in \mathbb{R}^{q\times q}$. Then the optimal solution to Eq. (4.25) is given by

$$T_Q^* = U\Sigma V^T.$$

Proof. Assume that the optimal T_Q^* to Eq. (4.25) shares the same left and right singular vectors as $\hat{S}_Q$. Then the problem in Eq. (4.25) is reduced to the problem in Eq. (4.26). Thus, all that remains is to show that T_Q^* shares the same left and right singular vectors as $\hat{S}_Q$.

Denote the Lagrangian function [38] associated with Eq. (4.25) as

$$H(T_Q, \lambda) = \frac{1}{2}\|T_Q - \hat{S}_Q\|_F^2 + \lambda(\|T_Q\|_* - \tau).$$

Since **0** is strictly feasible in Eq. (4.25), i.e., $\|0\|_* < \tau$, the Slater's condition [38] is satisfied and strong duality holds in Eq. (4.25). Let $\lambda^* \geq 0$ be the optimal dual variable [38] in Eq. (4.25). Therefore,

$$\begin{aligned} T_Q^* &= \arg\min_{T_Q} H(T_Q, \lambda^*) \\ &= \arg\min_{T_Q} \frac{1}{2}\|T_Q - \hat{S}_Q\|_F^2 + \lambda^*\|T_Q\|_*. \end{aligned}$$

Let $T_Q^* = U_T\Sigma_T V_T^T \in \mathbb{R}^{d\times m}$ be the SVD of T_Q^* and $r =$ rank(T_Q^*), where $U_T \in \mathbb{R}^{d\times r}$ and $U_T \in \mathbb{R}^{m\times r}$ are columnwise orthonormal, and $\Sigma_T \in \mathbb{R}^{r\times r}$ is diagonal consisting of non-zero singular values on the main diagonal. It is known [73] that the subdifferentials of $\|T_Q\|_*$ at T_Q^* can be expressed as

$$\partial\|T_Q^*\|_* = \left\{U_T V_T^T + D : D \in \mathbb{R}^{d\times m},\, U_T^T D = 0,\, DV_T = 0,\, \|D\|_2 \leq 1\right\}. \tag{4.27}$$

On the other hand, we can verify that T_Q^* is optimal to Eq.(4.25) if and only if **0** is a subgradient of $H(T_Q, \lambda^*)$ at T_Q^*, that is,

$$\mathbf{0} \in \partial H(T_Q^*, \lambda^*) = T_Q^* - \hat{S}_Q + \lambda^*\partial\|T_Q^*\|_*. \tag{4.28}$$

Let $U_T^\perp \in \mathbb{R}^{d\times(d-m)}$ and $V_T^\perp \in \mathbb{R}^{m\times(m-r)}$ be the null space [49] of U_T and V_T, respectively. It follows from Eq. (4.27) that there exists a point $D_T = U_T^\perp\Sigma_d\left(V_T^\perp\right)^T$ such that $U_T V_T^T + D_T \in \partial\|T_Q^*\|_*$ satisfies Eq. (4.28), and $\Sigma_d \in \mathbb{R}^{(d-m)\times(m-r)}$ is diagonal consisting of the singular values of D_T on the main diagonal. It follows that

$$\begin{aligned} \hat{S}_Q &= T_Q^* + \lambda^*\left(U_T V_T^T + D_T\right) \\ &= U_T\Sigma_T V_T^T + \lambda^* U_T V_T^T + \lambda^* U_T^\perp \Sigma_d\left(V_T^\perp\right)^T \\ &= U_T\left(\Sigma_T + \lambda^* I\right)V_T + U_T^\perp\left(\lambda^*\Sigma_d\right)\left(V_T^\perp\right)^T \end{aligned}$$

corresponds to the SVD of $\hat{S}_Q$. This completes the proof of this theorem. □

Note that the optimization problem in Eq. (4.26) is convex, and can be solved via an algorithm similar to the one in [74] proposed for solving the Euclidean projection onto the ℓ_1 ball.

4.5 Algorithms and Convergence

We present two algorithms based on the projected gradient scheme in Section 4.3 for solving the constrained convex optimization problem in Eq. (4.6), and analyze their rates of convergence using techniques in [45, 57].

Projected Gradient Algorithm

We first present a simple projected gradient algorithm. Let T_k be the feasible solution point in the k-th iteration; the projected gradient algorithm refines T_k by recycling the following two steps: find a candidate $\hat{T}$ for the subsequent feasible solution point T_{k+1} via

$$\hat{T} = T_{L,T_k} = \arg\min_{T \in \mathcal{M}} G_L(T_k, T),$$

and meanwhile ensure the step size $\frac{1}{L}$ satisfying the condition

$$F(\hat{T}) \leq G_L(T_k, \hat{T}).$$

Note that both T_k and $\hat{T}$ are feasible in Eq. (4.6). It follows from Eq. (4.20) that the solution sequence generated in the projected gradient algorithm leads to a non-increasing objective value in Eq. (4.6), that is,

$$F(T_{k-1}) \geq F(T_k), \ \forall k. \tag{4.29}$$

The pseudo-code of the projected gradient algorithm is presented in Algorithm 7, and its convergence rate analysis is summarized in Theorem 4.5.1. Note that the stopping criterion in line 11 of

```
 1: Input: $T_0$, $L_0 \in \mathbb{R}$, and max-iter.
 2: Output: $T$.
 3: for $i = 0, 1, \cdots$, max-iter do
 4:     while (true)
 5:         Compute $\hat{T} = T_{L_i,T_i}$ via Eq. (4.11).
 6:         if $F(\hat{T}) \leq G_{L_i}(T_i, \hat{T})$ then exit the loop.
 7:             else update $L_i = L_i \times 2$.
 8:         end-if
 9:     end-while
10:     Update $T_{i+1} = \hat{T}$ and $L_{i+1} = L_i$.
11:     if the stopping criterion is satisfied then exit the loop.
12: end-for
13: Set $T = T_{i+1}$.
```

Algorithm 7: Projected Gradient Method

Algorithm 7 can be set as: the change of objective values in two successive steps are smaller than some pre-specified value (e.g., 10^{-5}).

Theorem 4.5.1. *Let T^* be the global minimizer of Eq. (4.6); let $\hat{L}_f$ be the best Lipschitz continuous gradient defined in Eq.(4.7). Denote by k the index of iteration, and by T_k the solution point in the kth iteration of Algorithm 7. Then we have*

$$F(T_k) - F(T^*) \leq \frac{\hat{L}}{2k}\|T_0 - T^*\|_F^2,$$

where $\hat{L} = \max\{L_0, 2\hat{L}_f\}$, and L_0 and T_0 are the initial values of L_k and T_k in Algorithm 7, respectively.

Proof. It follows from Eq. (4.12) we have

$$T_{i+1} = T_{L_i,T_i} = T_i - \frac{1}{L_i}\mathcal{P}_{L_i}(T_i).$$

Moreover, from Eq. (4.16), we have

$$\begin{aligned} -\varepsilon_{i+1} &\geq \langle T^* - T_i, \mathcal{P}_{L_i}(T_i)\rangle + \frac{1}{2L_i}\|\mathcal{P}_{L_i}(T_i)\|_F^2 \\ &= \frac{L_i}{2}\left(-\|T_i\|_F^2 + \|T_{i+1}\|_F^2 + 2\langle T^*, T_i - T_{i+1}\rangle\right), \end{aligned} \tag{4.30}$$

where $\varepsilon_{i+1} = F(T_{i+1}) - F(T^*)$. Moving $L_i/2$ to the left side in Eq. (4.30) and summing such a reformulation from $i = 0$ to $i = k$, we have

$$\begin{aligned} \sum_{i=0}^{k} \frac{2}{L_i}\varepsilon_{i+1} &\leq \|T_0\|_F^2 - \|T_{k+1}\|_F^2 + 2\langle T^*, T_{k+1} - T_0\rangle \\ &= \|T_0 - T^*\|_F^2 - \|T_{k+1} - T^*\|_F^2 \\ &\leq \|T_0 - T^*\|_F^2. \end{aligned}$$

Since $L_i \geq L_{i-1}$ from line 7 in algorithm 7, and $\varepsilon_i \leq \varepsilon_{i-1}$ from Eq. (4.29) for all i, we have

$$\varepsilon_{k+1} \leq \frac{L_k}{2(k+1)}\|T_0 - T^*\|_F^2.$$

Moreover, it can be verified that $L_0 \leq L_k \leq 2\hat{L}_f$ for all k. This completes the proof of this theorem. □

Accelerated Projected Gradient Algorithm

The proposed projected gradient method Section 4.5 is simple to implement but converges slowly. We improve the projected gradient method using a scheme developed by Nesterov [45], which has been applied for solving various sparse learning formulations [75].

```
Input: $T_0$, $L_0 \in \mathbb{R}$, and max-iter.
Output: $T$.
Set $T_1 = T_0$, $t_{-1} = 0$, and $t_0 = 1$.
for $i = 1, 2, \cdots$, max-iter do
    Compute $\alpha_i = (t_{i-2} - 1)/t_{i-1}$.
    Compute $S = (1 + \alpha_i)T_i - \alpha_i T_{i-1}$.
    while (true)
        Compute $\hat{T} = T_{L_i,S}$ via Eq. (4.11).
        if $F(\hat{T}) \leq G_{L_i}(S, \hat{T})$ then exit the loop
            else update $L_i = L_i \times 2$.
        end-if
    end-while
    Update $T_{i+1} = \hat{T}$ and $L_{i+1} = L_i$.
    if the stopping criterion is satisfied then exit the loop.
    Update $t_i = \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$.
end-for
Set $T = T_{i+1}$.
```

Algorithm 8: Accelerated Projected Gradient Method

We utilize two sequences of variables in the accelerated projected gradient algorithm: (feasible) solution sequence $\{T_k\}$ and searching point sequence $\{S_k\}$. In the i-th iteration, we construct the searching point as

$$S_k = (1 + \alpha_k)T_k - \alpha_k T_{k-1}, \tag{4.31}$$

where the parameter $\alpha_k > 0$ is appropriately specified as shown in Algorithm 8. Similar to the projected gradient method, we refine the feasible solution point T_{k+1} via the general step as:

$$\hat{T} = T_{L,S_k} = \arg\min_{T \in \mathcal{M}} G_L(S_k, T),$$

and meanwhile determine the step size by ensuring

$$F(\hat{T}) \leq G_L(S_k, \hat{T}).$$

The searching point S_k may not be feasible in Eq. (4.6), which can be seen as a forecast of the next feasible solution point and hence leads to the faster convergence rate in Algorithm 8. The pseudo-code of the accelerated projected gradient algorithm is presented in Algorithm 8, and its convergence rate analysis is summarized in the following theorem.

Theorem 4.5.2. *Let T^* be the global minimizer of Eq. (4.6); let $\hat{L}_f$ be the best Lipschitz continuous gradient defined in Eq.(4.7). Denote by k the index of iteration, and by T_k the solution point in the kth iteration of Algorithm 8. Then we have*

$$F(T_{k+1}) - F(T^*) \leq \frac{2\hat{L}}{k^2} \|T_0 - T^*\|_F^2,$$

where $\hat{L} = \max\{L_0, 2\hat{L}_f\}$, where L_0 and T_0 are the initial values of L_k and T_k in Algorithm 8.

Proof. Denote $\varepsilon_i = F(T_i) - F(T^*)$. Setting $T = T_i$, $S = S_i$, and $L = L_i$ in Eq. (4.16), we have

$$\epsilon_i - \epsilon_{i+1} \geq \langle T_i - S_i, \mathcal{P}_{L_i}(S_i) \rangle + \frac{1}{2L_i} \| \mathcal{P}_{L_i}(S_i) \|_F^2, \tag{4.32}$$

where the left side of the inequality above follows from

$$T_{i+1} = T_{L_i,S_i} = \arg\min_{T \in \mathcal{M}} G_{L_i}(S_i, T_i).$$

Similarly, setting $T = T^*$, $S = S_i$, and $L = L_i$ in Eq. (4.16), we have

$$-\epsilon_{i+1} \geq \langle T^* - S_i, \mathcal{P}_{L_i}(S_i) \rangle + \frac{1}{2L_i} \| \mathcal{P}_{L_i}(S_i) \|_F^2. \tag{4.33}$$

Multiplying Eq. (4.32) by $t_{i-1} - 1$ and summing it with Eq. (4.33), we have

$$(t_{i-1} - 1)\,\varepsilon_i - t_{i-1}\varepsilon_{i+1} \geq \langle (t_{i-1} - 1)(T_i - S_i) + T^* - S_i, \mathcal{P}_{L_i}(S_i) \rangle + \frac{t_{i-1}}{2L_i} \| \mathcal{P}_{L_i}(S_i) \|_F^2. \tag{4.34}$$

Moreover, multiplying Eq. (4.34) by t_{i-1}, we have

$$t_{i-2}^2 \varepsilon_i - t_{i-1}^2 \varepsilon_{i+1} \geq \frac{1}{2L_i} \| t_{i-1} \mathcal{P}_{L_i}(S_i) \|_F^2 + \langle t_{i-1} \mathcal{P}_{L_i}(S_i), (t_{i-1} - 1)(T_i - S_i) + T^* - S_i \rangle. \tag{4.35}$$

where the left side is obtained via the equation

$$t_{i-1}^2 - t_{i-1} = t_{i-2}^2$$

from the line 15 in Algorithm 8. On the other hand, it follows from Eq. (4.12) we have

$$\mathcal{P}_{L_i}(S_i) = L_i \left(S_i - T_{L_i,S_i} \right) = L_i \left(S_i - T_{i+1} \right). \tag{4.36}$$

From Eq. (4.31) and the line 5 in Algorithm 8, we have

$$t_{i-1} S_i = t_{i-1} T_i + (t_{i-2} - 1)(T_i - T_{i-1}). \tag{4.37}$$

Denote

$$C_{i-2} = t_{i-2} T_i - (t_{i-2} - 1) T_{i-1} - T^*. \tag{4.38}$$

From Eqs. (4.36), (4.37) and (5.26), we can verify that

$$t_{i-1} \mathcal{P}_{L_i}(S_i) = t_{i-1} L_i (S_i - T_{i+1}) = L_i (C_{i-2} - C_{i-1}). \tag{4.39}$$

Moreover, we have

$$\begin{aligned} & (t_{i-1} - 1)(T_i - S_i) + T^* - S_i \\ = \;& (t_{i-1} - 1) T_i + T^* - t_{i-1} S_i \\ = \;& -t_{i-2} T_i + (t_{i-2} - 1) T_{i-1} + T^* = -C_{i-2}. \end{aligned} \tag{4.40}$$

Substituting Eqs. (4.39) and (4.40) into Eq. (4.35), we obtain

$$\begin{aligned} \|C_{i-1}\|_F^2 - \|C_{i-2}\|_F^2 &\leq \frac{2}{L_i}\left(t_{i-2}^2\varepsilon_i - t_{i-1}^2\varepsilon_{i+1}\right) \\ &\leq \frac{2}{L_{i-1}}t_{i-2}^2\varepsilon_i - \frac{2}{L_i}t_{i-1}^2\varepsilon_{i+1}. \end{aligned} \tag{4.41}$$

Summing Eq. (4.41) from $i = 1$ to $i = k$, we have

$$\|C_{k-1}\|_F^2 - \|C_{-1}\|_F^2 \leq \frac{2}{L_0}t_{-1}^2\varepsilon_1 - \frac{2}{L_k}t_{k-1}^2\varepsilon_{k+1}.$$

Therefore, we have

$$\begin{aligned} \frac{2}{L_k}t_{k-1}^2\varepsilon_{k+1} &\leq \|C_{-1}\|_F^2 - \|C_{k-1}\|_F^2 + \frac{2}{L_0}t_{-1}^2\varepsilon_1 \\ &\leq \|C_{-1}\|_F^2 + \frac{2}{L_0}t_{-1}^2\varepsilon_1 = \|T_0 - T^*\|^2, \end{aligned} \tag{4.42}$$

where the equality follows from $t_{-1} = 0$ in Algorithm 8. From line 15 in Algorithm 8, we have

$$2t_i = 1 + \sqrt{1 + 4t_{i-1}^2} \geq 2t_{i-1} + 1. \tag{4.43}$$

Summing Eq. (4.43) from $i = 1$ to $i = k$, we have

$$t_k \geq \frac{1}{2}(k+1), \quad \forall k. \tag{4.44}$$

Substituting Eq. (4.44) into Eq. (4.42), we complete the proof. □

The proof of Theorem 4.5.2 uses standard techniques in [45, 57] yet with simplification in several aspects for easy understanding. Note that the convergence rate achieved by Algorithm 8 is optimal among the first-order methods [45, 57].

4.6 Example: Learning Sparse and Low-Rank Patterns with Least Squares Loss

In this section, we present a concrete example of learning the sparse and low-rank patterns from multiple tasks, i.e., the MTL formulation in Eq. (4.5) using the least squares loss function; we also illustrate the use of the projected gradient algorithm (PG) and the accelerated projected gradient algorithm (AG) in this case. Mathematically, the specific MTL formulation can be expressed as

$$\begin{aligned} \min_{P,Q} \quad & \|(P+Q)^T X - Y\|_F^2 + \gamma\|P\|_1 \\ \text{subject to} \quad & \|Q\|_* \leq \tau, \end{aligned} \tag{4.45}$$

where $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$, and $Y = [y_1, y_2, \cdots, y_n] \in \mathbb{R}^{m\times n}$. For simplicity in Eq. (4.45) we assume that all of the m tasks share the same set of training data, and the derivation below can be easily extended to the case where each learning task has a different set of training data.

Efficient Computation for the Key Component

The computation of Eq. (4.11) is involved in each iteration of the projected gradient scheme. For the specifical MTL formulation in Eq. (4.45), given the intermediate solution pair $\{P_i, Q_i\}$ in the i-th iteration, the subsequent solution pair $\{P_{i+1}, Q_{i+1}\}$ can be obtained via

$$\begin{array}{ll} \min\limits_{\hat{P},\hat{Q}} & \frac{L_i}{2}\left\|\hat{P}-\tilde{P}_i\right\|_F^2 + \frac{L_i}{2}\left\|\hat{Q}-\tilde{Q}_i\right\|_F^2 + \gamma\|\hat{P}\|_1 \\ \text{subject to} & \|\hat{Q}\|_* \leq \tau, \end{array} \tag{4.46}$$

where L_i specifies the step size of the i-th iteration. The optimal $\hat{P}$ and $\hat{Q}$ to Eq. (4.46) can be obtained via solving two separate problems as below.

Computation of $\hat{P}$ The optimal $\hat{P}$ can be obtained via solving

$$\min_{\hat{P}} \quad \frac{L_i}{2}\left\|\hat{P}-\tilde{P}_i\right\|_F^2 + \gamma\|\hat{P}\|_1. \tag{4.47}$$

Based on the results in Section 4.4, the optimization problem in Eq. (4.47) can be further decomposed into entry-wise subproblems in the form of Eq. (4.23), which admits an analytical solution (Lemma 4.4.1).

Computation of $\hat{Q}$ The optimal $\hat{Q}$ can be obtained via solving

$$\begin{array}{ll} \min\limits_{\hat{Q}} & \left\|\hat{Q}-\tilde{Q}_i\right\|_F^2 \\ \text{subject to} & \|\hat{Q}\|_* \leq \tau. \end{array} \tag{4.48}$$

Based on the results in Section 4.4, the optimal solution to Eq. (4.48) can be obtained via the following two steps:

- Compute the SVD of $\tilde{Q}_i = U_{Q_i}\Sigma_{Q_i}V_{Q_i}^T$, where rank$(\tilde{Q}_i) = q$, $U_{Q_i} \in \mathbb{R}^{d\times q}$, $V_{Q_i} \in \mathbb{R}^{m\times q}$, and $\Sigma_{Q_i} =$ diag$(\hat{\varsigma}_1, \cdots \hat{\varsigma}_q) \in \mathbb{R}^{q\times q}$.
- Compute the optimal solution $\{\sigma_i^*\}_{i=1}^q$ to the following problem

$$\begin{array}{ll} \min\limits_{\{\sigma_i\}_{i=1}^q} & \sum_{i=1}^q (\sigma_i - \hat{\varsigma}_i)^2 \\ \text{subject to} & \sum_{i=1}^q \sigma_i \leq \tau,\ \sigma_i \geq 0. \end{array}$$

The optimal $\hat{Q}$ can be constructed as $\hat{Q} = U_{Q_i}\Sigma_Q V_{Q_i}^T$, where $\Sigma_Q =$ diag$(\sigma_1^*, \cdots \sigma_q^*)$.

Estimation of the Lipschitz Constant

An appropriate step size $1/L$ in Eq. (4.13) is important for the global convergence of the projected gradient based algorithms and its value can be estimated via many sophisticated line search schemes [38] in general. In Algorithm 7 (line $6 \sim 7$) and Algorithm 8 (line $9 \sim 10$), the value of L is updated until the inequality in Eq. (4.15) is satisfied; however, this updating procedure may incur overhead cost in the computation.

Denote the smooth component of the objective function in Eq. (4.45) by

$$f(P,Q) = \|(P+Q)^T X - Y\|_F^2. \tag{4.49}$$

It can be verified that any Lipschitz constant L_f of the function $f(P,Q)$ can satisfy Eq. (4.15). Note that the gradient of $f(P,Q)$ with respect to P and Q can be expressed as

$$\nabla_P f(P,Q) = \nabla_Q f(P,Q) = 2\left(XX^T(P+Q) - XY^T\right).$$

To avoid the computational cost of estimating the lipschitz constant for $f(P,Q)$, we directly estimate its best value (the smallest lipschitz constant), as summarized in the following lemma.

Lemma 4.6.1. *Given $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{m\times n}$, the best Lipschitz constant $\hat{L}_f$ of the function $f(P,Q)$ in Eq. (4.49) is no larger than $2\,\sigma_X^2$, where σ_X denotes the largest singular value of X.*

Proof. For arbitrary $P_x, P_y, Q \in \mathbb{R}^{d\times m}$, we have

$$\begin{aligned} \hat{L}_P = \frac{\|\nabla_{P_x} f(P_x,Q) - \nabla_{P_y} f(P_y,Q)\|_F}{\|P_x - P_y\|_F} &= \frac{\|2XX^T(P_x-P_y)\|_F}{\|P_x-P_y\|_F} \\ &\le \frac{2\,\sigma_X^2\|(P_x-P_y)\|_F}{\|P_x-P_y\|_F} = 2\,\sigma_X^2. \end{aligned} \tag{4.50}$$

Similarly, for arbitrary $P, Q_x, Q_y \in \mathbb{R}^{d\times m}$, we have

$$\hat{L}_Q = \frac{\|\nabla_{Q_x} f(P,Q_x) - \nabla_{Q_y} f(P,Q_y)\|_F}{\|Q_x - Q_y\|_F} \le 2\,\sigma_X^2. \tag{4.51}$$

Therefore it follows from Eq. (4.7) that

$$\hat{L}_f \le \max\left(\hat{L}_P, \hat{L}_Q\right) = 2\,\sigma_X^2. \tag{4.52}$$

This completes the proof. □

```
1:  Input: $P_0, Q_0, L = 2\,\sigma_X^2$, and max-iter.
2:  Output: $P, Q$.
3:  for $i = 0, 1, \cdots$ ,max-iter do
4:      Set $L_i = L, S_{P_i} = P_i, S_{Q_i} = Q_i$.
5:      Compute $\tilde{P}_i = S_{P_i} - \nabla_P f(P, Q)\,\big|_{P=S_{P_i}, Q=S_{Q_i}}$ ,
6:              $\tilde{Q}_i = S_{Q_i} - \nabla_Q f(P, Q)\,\big|_{P=S_{P_i}, Q=S_{Q_i}}$ .
7:      Compute $\hat{P}$ via Eq. (4.47) and $\hat{Q}$ via Eq. (4.48).
8:      Set $P_{i+1} = \hat{P}, Q_{i+1} = \hat{Q}$.
9:      if the stopping criterion is satisfied then exit the loop.
10: end-for
11: Set $P = P_{i+1}, Q = Q_{i+1}$.
```

Algorithm 9: Projected Gradient Algorithm (PG) for Solving Eq. (4.45)

```
1:  Input: $P_0, Q_0, L = 2\,\sigma_X^2$, and max-iter.
2:  Output: $P, Q$.
3:  Set $P_1 = P_0, Q_1 = Q_0$, $t_{-1} = 0$, and $t_0 = 1$.
4:  for $i = 1, 2, \cdots$ ,max-iter do
5:      Compute $\alpha_i = (t_{i-2} - 1)/t_{i-1}$.
6:      Set $L_i = L, S_{P_i} = (1 + \alpha_i)P_i - \alpha_i P_{i-1}, S_{Q_i} = (1 + \alpha_i)Q_i - \alpha_i Q_{i-1}$.
7:      Compute $\tilde{P}_i = S_{P_i} - \nabla_P f(P, Q)\,\big|_{P=S_{P_i}, Q=S_{Q_i}}$ ,
8:              $\tilde{Q}_i = S_{Q_i} - \nabla_Q f(P, Q)\,\big|_{P=S_{P_i}, Q=S_{Q_i}}$ .
9:      Compute $\hat{P}$ via Eq. (4.47), and $\hat{Q}$ via Eq. (4.48).
10:     Set $P_{i+1} = \hat{P}, Q_{i+1} = \hat{Q}$.
11:     if the stopping criterion is satisfied then exit the loop.
12:     Update $t_i = \frac{1}{2}(1 + \sqrt{1 + 4t_{i-1}^2})$.
13: end-for
14: Set $P = P_{i+1}, Q = Q_{i+1}$.
```

Algorithm 10: Accelerated Projected Gradient Algorithm (AG) for Solving Eq. (4.45)

Main Algorithms

The pseudo-codes of the PG and AG algorithms for solving Eq. (4.45) are presented in Algorithm 9 and Algorithm 10 respectively. The main difference between PG and AG lies in the construction of S_{P_i} and S_{Q_i}: in line 4 of Algorithm 9, S_{P_i} and S_{Q_i} are set as the pair of feasible points from the previous iteration; in line 6 of Algorithm 10, S_{P_i} and S_{Q_i} are set as the a linear combination of the feasible points from the previous and the current iterations, which are not necessary feasible in Eq. (4.45). The different construction leads to significant different rates of convergence, i.e., $\mathcal{O}(\frac{1}{k})$ in Algorithm 9 and $\mathcal{O}(\frac{1}{k^2})$ in Algorithm 10.

4.7 Empirical Evaluations

In this section, we evaluate the proposed multi-task learning formulation in comparison with other representative ones; we also conduct numerical studies on the projected gradient based algorithms.

Table 4.1: Statistics of the benchmark data sets.

Data Set	Sample Size	Dimension	Label	Type
Face	1400	19800	30	image
Scene	2407	294	6	image
Yeast	2417	103	14	gene
MediaMill$_1$	8000	120	80	multimedia
MediaMill$_2$	8000	120	100	multimedia
References	7929	26397	15	text
Science	6345	24002	22	text

We employ six benchmark data sets in our experiments. One of them is *AR Face Data* [76]: we use its subset consisting of 1400 face images corresponding to 100 persons. The other three are LIBSVM multi-label data sets[1]: for *Scene* and *Yeast*, we use the entire data sets; for *MediaMill*, we generate several subsets by randomly sampling 8000 data points with different numbers of labels. *References* and *Science* are Yahoo webpages data sets [77]: we preprocess the data sets following the same procedures in [78]. All of the benchmark data sets are normalized and their statistics are summarized in Table 4.1. Note that in our multi-task learning setting, each task corresponds to a label and we employ the least squares loss function for the following empirical studies.

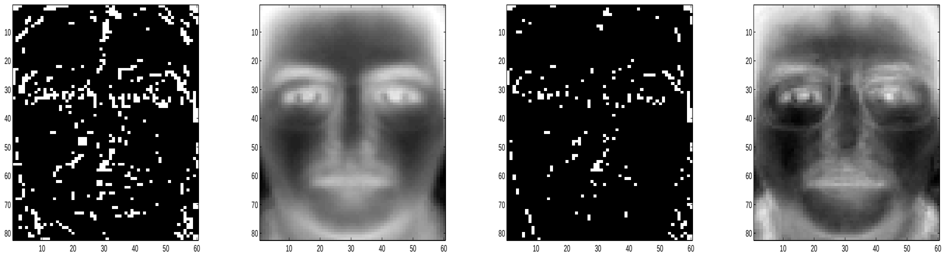


Figure 4.2: Extracted sparse (first and third plots) and low-rank (second and fourth plots) structures on AR face images with different sparse regularization and rank constraint parameters in Eq. (4.5): for the first two plots, we set $\gamma = 11, \tau = 0.08$; for the last two plots, we set $\gamma = 14, \tau = 0.15$.

Demonstration of Extracted Structures

We apply the proposed multi-task learning algorithm on the face images and then demonstrate the extracted sparse and low-rank structures. We use a subset of *AR Face Data* for this experiment. The original size of these images is 165×120; we reduce the size to 82×60.

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/multilabel/`

We convert the face recognition problem into the multi-task learning setting, where one task corresponds to learning a linear classifier, i.e., $f_\ell(x) = (p_\ell + q_\ell)^T x$, for recognizing the faces of one person. By solving Eq. (4.5), we obtained p_ℓ (sparse structure) and q_ℓ (low-rank structure); we reshape p_ℓ and q_ℓ and plot them in Figure 5.1. We only plot p_1 and q_1 for demonstration. The first two plots in Figure 5.1 are obtained by setting $\gamma = 11, \tau = 0.08$ in Eq. (4.5): we obtain a sparse structure of 15.07% nonzero entries and a low-rank structure of rank 3; similarly, the last two plots are obtained by setting $\gamma = 14, \tau = 0.15$, we obtain a sparse structure of 5.35% nonzero entries and a low-rank structure of rank 7. We observe that the sparse structure identifies the important detailed facial marks, and the low-rank structure preserves the rough shape of the human face; we also observe that a large sparse regularization parameter leads to high sparsity (lower percentage of the non-zero entries) and a large rank constraint leads to structures of high rank.

Performance Evaluation

We compare the proposed multi-task learning formulation with other representative ones in terms of average Area Under the Curve (AUC), Macro F1, and Micro F1 [79]. The reported experimental results are averaged over five random repetitions of the data sets into training and test sets of the ratio $1:9$. In this experiment, we stop the iterative procedure of the algorithms if the change of the objective values in two consecutive iterations is smaller than 10^{-5} or the iteration numbers larger than 10^5. The experimental setup is summarized as follows:

1. **MixedNorm**: The proposed multi-task learning formulation with the least squares loss. The trace-norm constraint parameter is tuned in $\{10^{-2} \times i\}_{i=1}^{10} \cup \{10^{-1} \times i\}_{i=2}^{10} \cup \{2 \times i\}_{i=1}^{p}$, where $p = \llcorner k/2 \lrcorner$ and k is the label number; the one-norm regularization parameter is tuned in $\{10^{-3} \times i\}_{i=1}^{10} \cup \{10^{-2} \times i\}_{i=2}^{10} \cup \{10^{-1} \times i\}_{i=2}^{10} \cup \{2 \times i\}_{i=1}^{10} \cup \{40 \times i\}_{i=1}^{20}$.

2. **OneNorm**: The formulation of the least squares loss with the one-norm regularization. The one-norm regularization parameter is tuned in $\{10^{-3} \times i\}_{i=1}^{10} \cup \{10^{-2} \times i\}_{i=2}^{10} \cup \{10^{-1} \times i\}_{i=2}^{10} \cup \{2 \times i\}_{i=1}^{10} \cup \{40 \times i\}_{i=1}^{20}$.

3. **TraceNorm**: The formulation of the least squares loss with the trace-norm constraint. The trace-norm constraint parameter is tuned in $\{10^{-2} \times i\}_{i=1}^{10} \cup \{10^{-1} \times i\}_{i=2}^{10} \cup \{2 \times i\}_{i=1}^{p}$, where $p = \llcorner k/2 \lrcorner$ and k denotes the label number.

4. **ASO**: The alternating structure optimization algorithm [16]. The regularization parameter is tuned in $\{10^{-3} \times i\}_{i=1}^{10} \cup \{10^{-2} \times i\}_{i=2}^{10} \cup \{10^{-1} \times 2\}_{i=1}^{10} \cup \{2 \times i\}_{i=1}^{10} \cup \{40 \times i\}_{i=1}^{20}$; the dimensionality of the shared subspace is tuned in $\{2 \times i\}_{i=1}^{p}$, where $p = \llcorner k/2 \lrcorner$ and k denotes the label number.

5. **IndSVM**: Independent support vector machines. The regularization parameter is tuned in $\{10^{-i}\}_{i=1}^{3} \cup \{2 \times i\}_{i=1}^{50} \cup \{200 \times i\}_{i=1}^{20}$.

6. **RidgeReg**: Ridge regression. The regularization parameter is tuned in $\{10^{-3} \times i\}_{i=1}^{10} \cup \{10^{-2} \times i\}_{i=2}^{10} \cup \{10^{-1} \times 2\}_{i=1}^{10} \cup \{2 \times i\}_{i=1}^{10} \cup \{40 \times i\}_{i=1}^{20}$.

The averaged performance (with standard deviation) of the competing algorithms are presented in Table 4.2 and Table 4.3. We have the following observations: (1) MixedNorm achieves the best performance among the competing algorithms on all benchmark data sets in this experiment, which gives strong support for our rationale of improving the generalization performance by learning the sparse and low-rank patterns simultaneously from multiple tasks; (2) TraceNorm outperforms OneNorm on *Scene* and *Yeast* data sets, which implies that the shared low-rank structure may be important for image and gene classification tasks; meanwhile, OneNorm outperforms TraceNorm on *MediaMill* and yahoo webpage data sets, which implies that sparse discriminative features may be important for multimedia learning problems; (3) the multi-task learning algorithms in our experiments outperform SVM and RidgeReg, which verifies the effect of improved generalization performance via multi-task learning.

Sensitivity Study

We conduct sensitivity studies on the proposed multi-task learning formulation, and study how the training ratio and the task number affect its generalization performance.

Effect of the training ratio We use *Scene* data for this experiment. We vary the training ratio in the set $\{0.1 \times i\}_{i=1}^{9}$ and record the obtained generalization performance for each training ratio. The experimental results are depicted in Figure 4.3. We can observe that (1) for all of the compared algorithms, the resulting generalization performance improves with the increase of the training ratio; (2) MixedNorm outperforms other competing algorithms in all cases in this experiment; (3) when the training ratio is small (e.g., smaller than 0.5), multi-task learning algorithms can significantly improve the generalization performance compared to IndSVM and RidgeReg; on the other hand, when the training ratio is large, all competing algorithms achieve comparable performance. This is consistent with previous observations that multi-task learning is most effective when the training size is small.

Effect of the task number We use *MediaMill* data for this experiment. We generate 5 data sets by randomly sampling 8000 data points with the task number set at $20, 40, 60, 80, 100$, respectively; for each data set, we set the training and test ratio at $1 : 9$ and record the average generalization per-

Table 4.2: Average performance (with standard derivation) comparison of six competing algorithms on three data sets in terms of average AUC (top section), Macro F1 (middle section), and Micro F1 (bottom section). All parameters of the six methods are tuned via cross-validation, and the reported performance is averaged over five random repetitions.

Data		*Scene*	*Yeast*	*References*
(n, d, m)		$(2407, 294, 6)$	$(2417, 103, 14)$	$(7929, 26397, 15)$
Average AUC	MixedNorm	91.602 ± 0.374	79.871 ± 0.438	77.526 ± 0.285
	OneNorm	87.846 ± 0.193	65.602 ± 0.842	75.444 ± 0.074
	TraceNorm	90.205 ± 0.374	76.877 ± 0.127	71.259 ± 0.129
	ASO	86.258 ± 0.981	64.519 ± 0.633	75.960 ± 0.104
	IndSVM	84.056 ± 0.010	64.601 ± 0.056	73.882 ± 0.244
	RidgeReg	85.209 ± 0.246	65.491 ± 1.160	74.781 ± 0.556
Macro F1	MixedNorm	60.602 ± 1.383	55.624 ± 0.621	37.135 ± 0.229
	OneNorm	55.061 ± 0.801	42.023 ± 0.120	36.579 ± 0.157
	TraceNorm	57.692 ± 0.480	52.400 ± 0.623	35.562 ± 0.278
	ASO	56.819 ± 0.214	45.599 ± 0.081	34.462 ± 0.315
	IndSVM	54.253 ± 0.078	38.507 ± 0.576	31.207 ± 0.416
	RidgeReg	53.281 ± 0.949	42.315 ± 0.625	32.724 ± 0.190
Micro F1	MixedNorm	64.392 ± 0.876	56.495 ± 0.190	59.408 ± 0.344
	OneNorm	59.951 ± 0.072	47.558 ± 1.695	58.798 ± 0.166
	TraceNorm	61.172 ± 0.838	54.172 ± 0.879	57.497 ± 0.130
	ASO	59.015 ± 0.124	45.952 ± 0.011	55.406 ± 0.198
	IndSVM	57.450 ± 0.322	52.094 ± 0.297	54.875 ± 0.185
	RidgeReg	56.012 ± 0.144	46.743 ± 0.625	53.713 ± 0.213

Table 4.3: Average performance (with standard derivation) comparison of six competing algorithms on three data sets in terms of average AUC (top section), Macro F1 (middle section), and Micro F1 (bottom section). All parameters of the six methods are tuned via cross-validation, and the reported performance is averaged over five random repetitions.

Data		*Science*	*MediaMill*$_1$	*MediaMill*$_2$
(n, d, m)		$(6345, 24002, 22)$	$(8000, 120, 80)$	$(8000, 120, 100)$
Average AUC	MixedNorm	75.746 ± 1.423	72.571 ± 0.363	65.932 ± 0.321
	OneNorm	74.456 ± 1.076	70.453 ± 0.762	64.219 ± 0.566
	TraceNorm	71.478 ± 0.293	69.469 ± 0.425	60.882 ± 1.239
	ASO	75.535 ± 1.591	71.067 ± 0.315	65.444 ± 0.424
	IndSVM	70.220 ± 0.065	67.088 ± 0.231	57.437 ± 0.594
	RidgeReg	69.177 ± 0.863	66.284 ± 0.482	56.605 ± 0.709
Macro F1	MixedNorm	38.281 ± 0.011	9.706 ± 0.229	7.981 ± 0.011
	OneNorm	37.981 ± 0.200	8.579 ± 0.157	6.447 ± 0.133
	TraceNorm	36.447 ± 0.055	8.562 ± 0.027	6.765 ± 0.039
	ASO	36.278 ± 0.183	8.023 ± 0.196	6.150 ± 0.023
	IndSVM	35.175 ± 0.177	6.207 ± 0.410	5.175 ± 0.177
	RidgeReg	35.066 ± 0.196	7.724 ± 0.190	5.066 ± 0.096
Micro F1	MixedNorm	52.619 ± 0.042	61.426 ± 0.062	60.117 ± 0.019
	OneNorm	52.733 ± 0.394	60.594 ± 0.026	59.221 ± 0.39
	TraceNorm	49.124 ± 0.409	59.090 ± 0.117	58.317 ± 1.01
	ASO	49.616 ± 0.406	59.415 ± 0.005	59.079 ± 1.72
	IndSVM	48.574 ± 0.265	57.825 ± 0.272	56.525 ± 0.317
	RidgeReg	47.454 ± 0.255	57.752 ± 0.210	56.982 ± 0.455

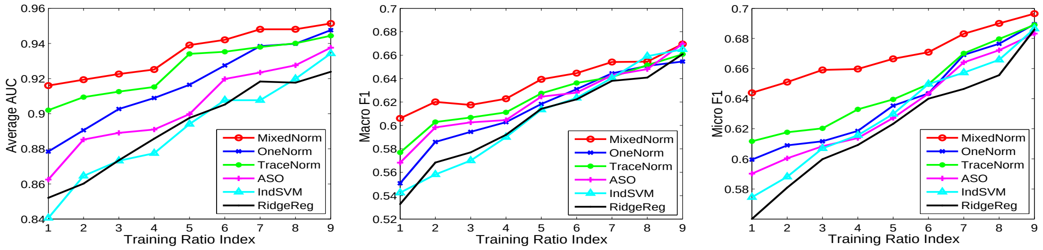


Figure 4.3: Performance comparison of six multi-task learning algorithms with different training ratios in terms of average AUC (left plot), Macro F1 (middle plot), and Micro F1 (right plot). The index on x-axis corresponds to the training ratio varying from 0.1 to 0.9.

formance of the multi-task learning algorithms over 5 random repetitions. The experimental results are depicted in Figure 4.4. We can observe that (1) for all of the compared algorithms, the achieved performance decreases with the increase of the task numbers; (2) MixedNorm outperforms or perform competitively compared to other algorithms with different task numbers; (3) all of the specific multi-task learning algorithms outperform IndSVM and RidgeReg. Note that the learning problem becomes more difficult as the number of the tasks increases, leading to decreased performance for both multi-task and single-task learning algorithms. We only present the performance comparison in terms of Macro/Micro F1; we observe a similar trend in terms of average AUC in the experiments.

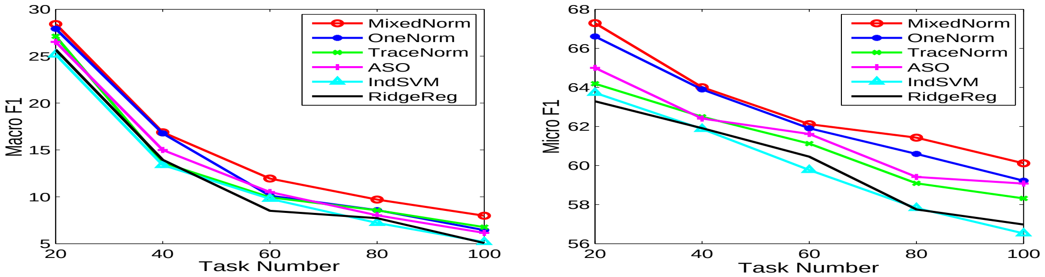


Figure 4.4: Performance comparison of the six competing multi-task learning algorithms with different numbers of tasks in terms of Macro F1 (top plot) and Micro F1 (bottom plot).

Comparison of PG and AG

We empirically compare the projected gradient algorithm (PG) in Algorithm 7 and the accelerated projected gradient algorithm (AG) in Algorithm 8 using *Scene* data. We present the comparison results of setting $\gamma = 1, \tau = 2$ and $\gamma = 6, \tau = 4$ in Eq. (4.5); for other parameter settings, we observe similar trends in our experiments.

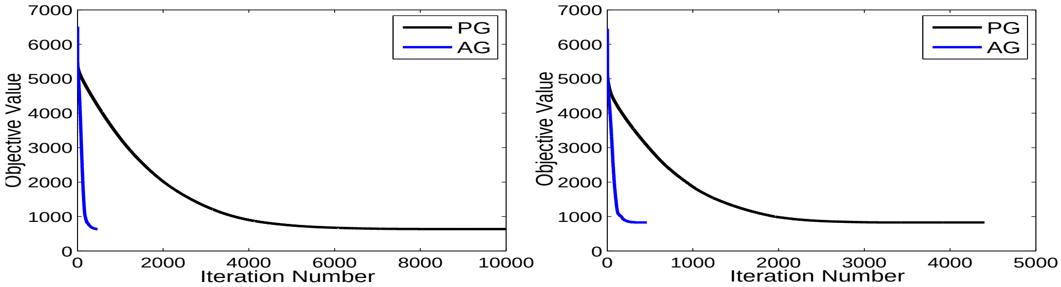


Figure 4.5: Convergence rate comparison between PG and AG: the relationship between the objective value of Eq. (4.5) and the iteration number (achieved via PG and AG, respectively). For the left plot, we set $\gamma = 1, \tau = 2$; for the right plot, we set $\gamma = 6, \tau = 4$.

Comparison on convergence rate We apply PG and AG for solving Eq. (4.5) respectively, and compare the relationship between the obtained objective values and the required iteration numbers. The experimental setup is as follows: we terminate the PG algorithm when the change of objective values in two successive steps is smaller than 10^{-5} and record the obtained objective value; we then use such a value as the stopping criterion in AG, that is, we stop AG when AG attains an objective value equal to or smaller than the one attained by PG. The experimental results are presented in Figure 4.5. We can observe that AG converges much faster than PG, and their respective convergence speeds are consistent with the theoretical convergence analysis in Section 6.3, that is, PG converges at the rate of $\mathcal{O}(1/k)$ and AG at the rate of $\mathcal{O}(1/k^2)$, respectively.

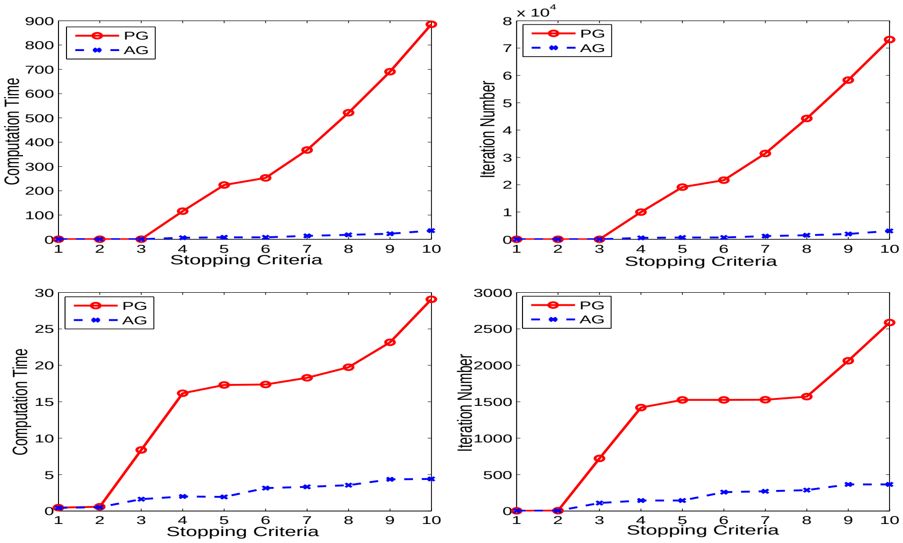


Figure 4.6: Comparison of PG and AG in terms of the computation time in seconds (left column) and iteration number (right column) with different stopping criteria. The x-axis indexes the stopping criterion from 10^{-1} to 10^{-10}. Note that we stop PG or AG when the change of the objective value in Eq. (4.5) is smaller than the value of the stopping criterion. For the first row, we set $\gamma = 1, \tau = 2$; for the second row, we set $\gamma = 6, \tau = 4$.

Comparison on computation cost We compare PG and AG in terms of computation time (in seconds) and iteration numbers (for attaining convergence) by using different stopping criteria $\{10^{-i}\}_{i=1}^{10}$. We stop PG and AG if the stopping criterion is satisfied, that is, the change of the objective values in two successive steps is smaller than 10^{-i}. The experimental results are presented in Table 4.4 and Figure 4.6. We can observe from these results that (1) PG and AG require higher computation costs (more computation time and larger numbers of iterations) for a smaller value of the stopping criterion (higher accuracy in the optimal solution); (2) in general, AG requires lower computation costs than PG in this experiment; such an efficiency improvement is more significant when a smaller value is used in the stopping criterion.

Table 4.4: Comparison of PG and AG in terms of computation time (in seconds) and iteration number using different stopping criteria.

	$\gamma = 1, \tau = 2$				$\gamma = 6, \tau = 4$			
stopping	iteration		time		iteration		time	
criteria	PG	AG	PG	AG	PG	AG	PG	AG
10^{-1}	2	2	0.6	0.4	3	3	0.5	0.4
10^{-2}	4	4	0.6	0.4	5	4	0.6	0.5
10^{-3}	17	15	0.6	0.5	722	110	8.4	1.6
10^{-4}	9957	537	116.1	6.5	1420	144	16.2	1.9
10^{-5}	19103	683	223.7	8.3	1525	144	17.3	1.9
10^{-6}	21664	683	253.0	8.3	1525	259	17.4	3.1
10^{-7}	31448	1199	367.9	14.3	1527	271	18.3	3.3
10^{-8}	44245	1491	521.3	18.4	1570	287	19.7	3.5
10^{-9}	58280	1965	690.5	23.0	2062	365	23.1	4.2
10^{-10}	73134	3072	885.4	35.9	2587	365	29.1	4.4

Automated Annotation of the Gene Expression Pattern Images

We apply the proposed multi-task learning formulation for the automated annotation of the *Drosophila* gene expression pattern images from the FlyExpress[2] database.

We preprocess the *Drosophila* gene expression pattern images (of the standard size 128×320) from the FlyExpress database following the procedures in [27]. The *Drosophila* images are from 16 specific stages, which are then grouped into 6 stage ranges ($1 \sim 3, 4 \sim 6, 7 \sim 8, 9 \sim 10, 11 \sim 12$, $13 \sim 16$). We manually annotate the image groups (based on the genes and the developmental stages) using the structured CV terms. Each image group is then represented as a feature vector based on the bag-of-words and the soft-assignment sparse coding. Note that the SIFT (scale-invariant feature transform) features [28] are extracted from the images with the patch size set at 16×16 and the number of visual words in sparse coding set at 2000. The first stage range only

[2] `http://www.flyexpress.net/`

Table 4.5: Performance comparison of six competing algorithms for the gene expression pattern images annotation (10 CV terms) in terms of average AUC (top section), Macro F1 (middle section), and Micro F1 (bottom section). All parameters of the six methods are tuned via cross-validation, and the reported performance is averaged over five random repetitions. Note that n, d, and m denote the sample size, dimensionality, and term (task) number, respectively.

Stage Range		$4 \sim 6$	$7 \sim 8$	$9 \sim 10$	$11 \sim 12$	$13 \sim 16$
(n, d, m)		$(925, 2000, 10)$	$(797, 2000, 10)$	$(919, 2000, 10)$	$(1622, 2000, 10)$	$(2228, 2000, 10)$
Avg. AUC	MixedNorm	75.44 ± 0.87	75.55 ± 0.42	77.18 ± 0.50	83.82 ± 0.93	85.54 ± 0.25
	OneNorm	74.98 ± 0.12	73.80 ± 0.55	75.80 ± 0.24	82.78 ± 0.27	84.77 ± 0.20
	TraceNorm	73.04 ± 0.79	74.06 ± 0.46	76.71 ± 0.72	81.77 ± 1.10	83.64 ± 0.27
	ASO	72.01 ± 0.36	73.56 ± 0.97	75.89 ± 0.24	82.97 ± 0.15	83.06 ± 0.80
	IndSVM	71.00 ± 0.53	72.13 ± 0.70	73.58 ± 0.48	79.01 ± 0.58	82.06 ± 1.04
	RidgeReg	72.46 ± 0.15	72.51 ± 0.82	73.10 ± 0.38	80.83 ± 0.67	82.02 ± 0.15
Mac. F1	MixedNorm	43.71 ± 0.32	48.31 ± 0.56	53.11 ± 0.56	61.11 ± 0.58	61.81 ± 0.40
	OneNorm	42.24 ± 0.14	47.40 ± 0.23	51.04 ± 0.10	59.36 ± 0.60	61.02 ± 0.10
	TraceNorm	41.38 ± 0.36	46.51 ± 0.67	51.13 ± 0.95	61.05 ± 0.78	60.15 ± 0.45
	ASO	42.13 ± 0.63	47.83 ± 1.55	51.18 ± 0.41	61.01 ± 0.55	60.58 ± 0.19
	IndSVM	40.88 ± 0.49	46.73 ± 0.51	50.28 ± 0.65	59.82 ± 0.83	59.62 ± 0.94
	RidgeReg	41.65 ± 0.45	46.91 ± 0.94	50.69 ± 0.77	59.46 ± 0.95	60.59 ± 0.79
Mic. F1	MixedNorm	46.98 ± 0.90	62.73 ± 0.93	63.46 ± 0.07	69.31 ± 0.37	67.13 ± 0.41
	OneNorm	44.55 ± 0.38	60.02 ± 0.56	61.78 ± 0.10	68.54 ± 0.17	66.30 ± 0.55
	TraceNorm	43.88 ± 0.73	61.29 ± 0.78	61.33 ± 1.04	68.68 ± 0.27	66.37 ± 0.26
	ASO	44.77 ± 0.49	60.47 ± 0.23	62.26 ± 0.23	68.60 ± 0.61	66.25 ± 0.18
	IndSVM	42.05 ± 0.61	60.09 ± 0.78	60.57 ± 0.75	67.08 ± 0.99	65.95 ± 0.80
	RidgeReg	43.63 ± 0.41	59.95 ± 0.75	60.59 ± 0.66	66.87 ± 0.11	65.67 ± 1.10

contains 2 CV terms and we do not report the performance for this stage range. For other stage ranges, we consider the top 10 and 20 CV terms that appears the most frequently in the image groups and treat the annotation of each CV term as one task. We generate 10 subsets for this experiment, and randomly partition each subset into training and test sets using the ratio $1 : 9$. Note that the parameters in the competing algorithms are tuned as the experimental setting in Section 4.7.

We report the averaged AUC (Avg. AUC), Macro F1 (Mac. F1), and Micro F1 (Mic. F1) over 10 random repetitions in Table 4.5 (for 10 CV terms) and Table 4.6 (for 20 CV terms), respectively. We can observe that MixedNorm achieves the best performance among the six algorithms on all subsets. In particular, MixedNorm outperforms the multi-task learning algorithms: OneNorm, TraceNorm, and ASO; MixedNorm also outperforms the single-task learning algorithms: IndSVM and RidgeReg. The experimental results demonstrate the effectiveness of learning the sparse and low-rank patterns from multiple tasks for improved generalization performance.

4.8 Summary

We consider the problem of learning sparse and low-rank patterns from multiple related tasks. We propose a multi-task learning formulation in which the sparse and low-rank patterns are induced respectively by a cardinality regularization term and a low-rank constraint. The proposed formula-

Table 4.6: Performance comparison of six competing algorithms for the gene expression pattern images annotation (20 CV terms).

Stage Range		$4 \sim 6$	$7 \sim 8$	$9 \sim 10$	$11 \sim 12$	$13 \sim 16$
(n, d, m)		$(1023, 2000, 20)$	$(827, 2000, 20)$	$(1015, 2000, 20)$	$(1940, 2000, 20)$	$(2476, 2000, 20)$
Avg. AUC	MixedNorm	76.27 ± 0.53	72.03 ± 0.63	73.97 ± 1.10	82.27 ± 0.42	82.16 ± 0.16
	OneNorm	75.13 ± 0.03	70.95 ± 0.14	72.49 ± 1.00	81.73 ± 0.36	81.03 ± 0.08
	TraceNorm	74.69 ± 0.39	69.43 ± 0.46	71.59 ± 0.79	81.53 ± 0.16	80.88 ± 1.10
	ASO	74.86 ± 0.33	70.15 ± 0.31	71.37 ± 0.99	81.45 ± 0.26	80.79 ± 0.23
	IndSVM	73.82 ± 0.78	69.74 ± 0.19	70.84 ± 0.85	80.86 ± 0.56	79.94 ± 0.19
	RidgeReg	74.66 ± 1.44	70.77 ± 0.62	69.36 ± 1.44	80.40 ± 0.43	78.29 ± 0.42
Mac. F1	MixedNorm	31.90 ± 0.11	31.13 ± 0.68	32.28 ± 1.13	43.48 ± 0.39	43.44 ± 0.60
	OneNorm	30.48 ± 0.12	30.07 ± 0.56	30.50 ± 1.13	41.89 ± 0.24	42.64 ± 0.47
	TraceNorm	29.22 ± 0.31	30.24 ± 0.78	31.28 ± 0.54	42.07 ± 0.67	41.11 ± 0.52
	ASO	30.51 ± 0.94	29.37 ± 0.56	31.46 ± 1.33	42.34 ± 1.08	41.55 ± 0.67
	IndSVM	29.47 ± 0.46	28.85 ± 0.62	30.03 ± 1.68	41.63 ± 0.58	40.80 ± 0.66
	RidgeReg	28.92 ± 1.24	28.76 ± 0.95	29.94 ± 1.84	41.51 ± 0.39	40.84 ± 0.40
Mic. F1	MixedNorm	42.50 ± 0.63	57.04 ± 0.13	57.37 ± 0.71	61.97 ± 0.51	56.75 ± 0.40
	OneNorm	40.80 ± 0.48	56.55 ± 0.22	56.82 ± 0.04	60.59 ± 0.32	55.87 ± 0.11
	TraceNorm	41.26 ± 1.16	56.47 ± 0.27	55.37 ± 0.38	59.27 ± 0.93	54.08 ± 0.51
	ASO	40.80 ± 0.53	56.88 ± 0.13	55.65 ± 0.33	59.74 ± 0.18	54.83 ± 0.67
	IndSVM	39.24 ± 0.82	55.40 ± 0.15	55.75 ± 1.70	58.33 ± 0.53	53.61 ± 0.36
	RidgeReg	38.46 ± 0.41	56.08 ± 0.46	54.23 ± 0.85	59.13 ± 0.67	53.75 ± 0.31

tion is non-convex; we convert it into its tightest convex surrogate and then propose to apply the general projected gradient scheme to solve such a convex surrogate. We present the procedures for computing the projected gradient and ensuring the global convergence of the projected gradient scheme. Moreover, we show that the projected gradient can be obtained via solving two simple convex subproblems. We also present two detailed projected gradient based algorithms and analyze their rates of convergence. Additionally, we illustrate the use of the presented projected gradient algorithms for the proposed multi-task learning formulation using the least squares loss. Our experiments demonstrate the effectiveness of the proposed multi-task learning formulation and the efficiency of the proposed projected gradient algorithms.

Chapter 5

Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning

5.1 Introduction

In many real-world applications involving multiple tasks, it is usually the case that a group of tasks are related while some other tasks are irrelevant to such a group. Simply pooling all tasks together and learning them simultaneously under a presumed structure may degrade the overall learning performance. It is thus desirable to identify irrelevant (outlier) tasks in the development of the multi-task learning algorithms. Learning multiple tasks under this setting is usually referred to as robust multi-task learning [80].

Recently robust multi-task learning has received increasing attention in the areas of data mining and machine learning. In [3, 14, 15], the task clustering (TC) approach is proposed for discovering the common structures in multiple learning tasks. The main idea behind the TC algorithms is to cluster similar tasks into different groups and constrain the tasks from the same group to share the same model representation or parameters. In [80, 81], multivariate student t-processes and their generalization are proposed for distinguishing good tasks from noisy or outlier tasks. The t-processes-based MTL algorithms model the relationship of multiple tasks using a task covariance matrix and they are robust by nature as t-process is implicitly an infinite Gaussian mixture. In [82, 83], the block-sparse structures ($\ell_{1,\infty}$-norm or $\ell_{2,1}$-nrom) are employed to extract essential features shared across the tasks and hence improve the robustness of the learning algorithms.

In this chapter, we propose a robust multi-task learning (RMTL) algorithm which learns multiple tasks simultaneously as well as identifies the irrelevant (outlier) tasks. Specifically, our proposed RMTL algorithm captures the relationship of multiple related tasks using a low-rank structure and meanwhile identifies the outlier tasks using a group-sparse structure. The proposed RMTL algorithm is formulated as a non-smooth convex (unconstrained) optimization problem in which the least squares loss is regularized by a nonnegative linear combination of the trace norm and the $\ell_{1,2}$-norm. The optimization problems involving the trace norm and the $\ell_{1,2}$-norm can be routinely reformulated as semi-definite programs or second-order cone programs, both of which, however, are not scalable to large-scale data. We propose to adopt the accelerated proximal method (APM) for solving the proposed RMTL formulation efficiently. One key component in applying APM for solving RMTL is the computation of the associated proximal operator, which is a non-smooth optimization problem involving two optimization variables. The associated proximal operator can be shown to admit an analytic solution. We also conduct theoretical analysis on the performance bound of the composite regularization in RMTL. We first present key properties of the optimal solution to RMTL (Lemma 6.4.4). We then present an assumption associated with the prescribed training samples and the geometric structures of the matrices of interest; based on this assumption, we derive a performance bound for the combined regularization for multi-task regression (Theorem 6.4.1). We conduct simulations on benchmark data sets to demonstrate the effectiveness and efficiency of the proposed algorithm.

5.2 Robust MTL Framework

Assume that we are given m (regression) learning tasks. Each task is associated with a set of training data

$$\{(x_1^i, y_1^i), \cdots, (x_{n_i}^i, y_{n_i}^i)\} \subset \mathbb{R}^d \times \mathbb{R},\ i \in \mathbb{N}_m, \tag{5.1}$$

and a linear predictive function f_i as

$$f_i(x_j^i) = w_i^T x_j^i \approx y_j^i,\ x_j^i \in \mathbb{R}^d,\ y_j^i \in \mathbb{R}, \tag{5.2}$$

where i and j index the task and the training sample respectively, w_i is the weight vector, n_i and d denote the training sample size and the feature dimensionality respectively.

We consider the multi-task learning setting where multiple tasks are divided into two groups, i.e., the related tasks group and the irrelevant (outlier) tasks group. We consider a composite structure which couples the related tasks using a low-rank structure and identifies the outlier tasks using a group-sparse structure. Denote the transformation matrix of the m tasks by $W = [w_1, \cdots, w_m] \in \mathbb{R}^{d\times m}$. Specifically, W is given by the direct summation of a low-rank matrix $L = [l_1, \cdots, l_m] \in \mathbb{R}^{d\times m}$ (of a smaller set of basis factors), and a group-sparse (column-sparse) matrix $S = [s_1, \cdots, s_m] \in \mathbb{R}^{d\times m}$ (of zero-vectors in the columns). The weight vector of the i-th task can be expressed as

$$w_i = l_i + s_i,\ l_i \in \mathbb{R}^d,\ s_i \in \mathbb{R}^d,\ i \in \mathbb{N}_m, \tag{5.3}$$

where l_i and s_i are from the aforementioned low-rank structure and the group-sparse structure, respectively.

We propose a robust multi-task learning formulation (RMTL) to learn multiple tasks simultaneously as well as identify the irrelevant outlier tasks. Mathematically, RMTL is formulated as

$$\min_{L,S}\ \mathcal{L}\left((l_i + s_i)^T x_j^i, y_j^i\right) + \alpha\|L\|_* + \beta\|S\|_{1,2}, \tag{5.4}$$

where the trace norm regularization term encourages the desirable low-rank structure in the matrix L (for coupling the related tasks), and the $\ell_{1,2}$-norm regularization term induces the desirable group-sparse structure in the matrix S (for identifying the outlier tasks), α and β are non-negative trade-off parameters, and $\mathcal{L}(\cdot,\cdot)$ represents the commonly used least squares loss function. Note that the empirical evaluation of the (averaged) least square loss of the m tasks over the prescribed training data can be expressed as

$$\mathcal{L}\left((l_i + s_i)^T x_j^i, y_j^i\right) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \frac{1}{mn_i} \left((l_i + s_i)^T x_j^i - y_j^i\right)^2 . \tag{5.5}$$

Our motivation behind the proposed RMTL formulation in Eq. (5.4) is as follows: if the i-th task is from the related tasks group, s_i is expected to be a zero-vector and hence w_i obeys the specified low-rank structure constraint; on the other hand, if the i-th task is from the outlier tasks group, s_i is expected to be non-zero and w_i is equal to a direct sum of l_i and the non-zero s_i.

The RMTL formulation in Eq. (5.4) is an unconstrained convex optimization problem with a non-smooth objective function. Such a problem is difficult to solve directly due to the non-smoothness in the trace norm and the $\ell_{1,2}$-norm regularization terms.

The proposed RMTL formulation in Eq. (5.4) subsumes several representative algorithms as special cases. As $\beta \to +\infty$, RMTL is degenerated into

$$\min_{L} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \frac{1}{mn_i} \left(l_i^T x_j^i - y_j^i\right)^2 + \alpha \|L\|_*. \tag{5.6}$$

The formulation in Eq. (5.6) is essentially the least squares regression with trace norm regularization, in which multiple learning tasks are coupled via a low-rank structure. On the other hand, as $\alpha \to \infty$, RMTL is degenerated into

$$\min_{S} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \frac{1}{mn_i} \left(s_i^T x_j^i - y_j^i\right)^2 + \beta \|S\|_{1,2}. \tag{5.7}$$

The formulation in Eq. (5.7) is essentially a variant of the ridge regression with the smooth term $\sum_{i=1}^{m} \|s_i\|^2$ replaced by the non-smooth term $\sum_{i=1}^{m} \|s_i\|$. In such a formulation, the multiple tasks are decoupled and each task can be learned (optimized) via

$$\min_{s_i} \frac{1}{mn_i} \sum_{j=1}^{n_i} \left(s_i^T x_j^i - y_j^i\right)^2 + \beta \|s_i\|_2.$$

Note that similar low-rank and group-sparse structures are studied from a different perspective in [84, 85], which focus on decomposing a given data matrix into a unique sum of a low-rank structure and a column-sparse structure and providing a theoretical guarantee for existence and uniqueness of the decomposition.

5.3 Accelerated Proximal Method

In this section, we consider to solve the RMTL formulation in Eq. (5.4) using the accelerated proximal method (APM) [45, 57, 71]. APM has attracted extensive attentions in the machine learning and data mining communities [75, 86–90] due to its optimal convergence rate among all first-order techniques and its ability of dealing with large-scale non-smooth optimization problems. Note that in this chapter, we focus on discussing the key ingredient of APM, i.e, the proximal operator and its efficient computation; the detailed description of APM can be found in [45, 57, 71].

Proximal Operator

For the optimization problem in Eq. (5.4), we symbolically denote its variables by

$$Z = \begin{bmatrix} L \\ S \end{bmatrix}, \ L \in \mathbb{R}^{d\times m}, \ S \in \mathbb{R}^{d\times m},$$

and denote the smooth and non-smooth components of its objective function respectively by

$$f(Z) = \mathcal{L}\left((l_i + s_i)^T x_j^i, y_j^i\right), \quad g(Z) = \alpha\|L\|_* + \beta\|S\|_{1,2}. \tag{5.8}$$

To solve Eq. (5.4), APM maintains two sequences of variables: a feasible solution sequence $\{Z_k\}$ and a searching point sequence $\{\widehat{Z}_k\}$. The general scheme of APM can be described as below: at the k-th iteration of APM, the solution point Z_{k+1} can be computed via

$$Z_{k+1} = \arg\min_Z \frac{\gamma_k}{2}\left\|Z - \left(\widehat{Z}_k - \frac{1}{\gamma_k}\nabla f(\widehat{Z}_k)\right)\right\|_F^2 + g(Z), \tag{5.9}$$

where $\widehat{Z}_k$ denotes a searching point constructed from a linear combination of Z_k and Z_{k-1} from previous iterations, and $\nabla f(\widehat{Z}_k)$ denotes the derivative of the smooth component $f(\cdot)$ in Eq. (5.8) at $\widehat{Z}_k$, γ_k specifies the step size which can be appropriately determined by iteratively increasing its value until the inequality

$$f(Z_{k+1}) \le f(\widehat{Z}_k) + \langle \nabla f(\widehat{Z}_k), Z_{k+1} - \widehat{Z}_k\rangle + \frac{\gamma_k}{2}\|Z_{k+1} - \widehat{Z}_k\|_F^2, \tag{5.10}$$

is satisfied. The procedure in Eq. (6.7) is commonly referred to as the proximal operator [91]. The efficient computation of the proximal operator is critical for the practical convergence of APM, as it is involved in each iteration of the APM algorithm.

For the optimization problem in Eq. (5.4), its proximal operator can be expressed as an optimization problem of the general form

$$\min_{L_z, S_z} \quad \|L_z - L_{\hat{z}}\|_F^2 + \|S_z - S_{\hat{z}}\|_F^2 + \hat{\alpha}\|L_z\|_* + \hat{\beta}\|S_z\|_{1,2}, \tag{5.11}$$

where $\hat{\alpha} = \frac{2\alpha}{\gamma_k}$ and $\hat{\beta} = \frac{2\beta}{\gamma_k}$. It can be easily verified that the optimization of L_z and S_z in Eq. (5.11) are decoupled. Moreover, the optimal solution to Eq. (5.11) admits an analytic form as presented below.

Computation of L_z The optimal L_z to Eq. (5.11) can be obtained by solving the following optimization problem:

$$\min_{L_z} \quad \|L_z - L_{\hat{z}}\|_F^2 + \hat{\alpha}\|L_z\|_*. \tag{5.12}$$

The computation procedure above is equal to the matrix shrinkage operator discussed in [92, 93]. In essence it applies soft-thresholding to the non-zero singular values [94] of $L_{\hat{z}}$ as summarized in the following theorem.

Theorem 5.3.1. *Given an arbitrary $L_{\hat{z}}$ in Eq. (5.12), let rank$(L_{\hat{z}}) = r$ and denote the singular value decomposition (SVD) of $L_{\hat{z}}$ in the reduced form as*

$$L_{\hat{z}} = U_{\hat{z}} \Sigma_{\hat{z}} V_{\hat{z}}^T, \;\; \Sigma_{\hat{z}} = \textit{diag}\left(\{\sigma_i\}_{i=1}^r\right)$$

where, $U_{\hat{z}} \in \mathbb{R}^{d \times r}$ and $V_{\hat{z}} \in \mathbb{R}^{m \times r}$ consist of orthonormal columns, $\Sigma_{\hat{z}} \in \mathbb{R}^{r \times r}$ is diagonal, and $\{\sigma_i\}_{i=1}^r$ represent the non-zero singular values. Then the optimal L_z^* to Eq. (5.12) is given by

$$L_z^* = U_{\hat{z}} \, \textit{diag}\left(\left\{\sigma_i - \frac{1}{2}\hat{\alpha}\right\}_+\right) V_{\hat{z}}^T,$$

where $\{e\}_+ = \max(e, 0)$.

The dominating cost in solving Eq. (5.12) lies in the compact SVD operation on the matrix $L_{\hat{z}} \in \mathbb{R}^{d \times m}$ ($m \ll d$ in general MTL settings).

Computation of S_z The optimal S_z to Eq. (5.11) can be obtained by solving the following optimization problem:

$$\min_{S_z} \quad \|S_z - S_{\hat{z}}\|_F^2 + \hat{\beta}\|S_z\|_{1,2}. \tag{5.13}$$

It can be easily verified that in Eq. (5.13) the column vectors of S_z can be optimized separately. Specifically, each vector of the optimal S_z to Eq. (5.13) can be obtained via solving a subproblem in the form

$$\min_{s} \quad \|s - \hat{s}\|_2^2 + \hat{\beta}\|s\|_2. \tag{5.14}$$

It can be verified that the optimization problem above admits an analytic solution [86] as summarized in the following lemma.

Lemma 5.3.1. *Let s^* be the optimal solution to the optimization problem in Eq. (5.14). Then s^* is given by*

$$s^* = \begin{cases} \hat{s}\left(1 - \frac{\hat{\beta}}{2\|\hat{s}\|_2}\right) & \|\hat{s}\|_2 > \frac{\hat{\beta}}{2} \\ 0 & 0 \leq \|\hat{s}\|_2 \leq \frac{\hat{\beta}}{2} \end{cases}.$$

Proof. Denote the objective function in Eq. (5.14) by $z(s)$ as

$$z(s) = \|s - \hat{s}\|_2^2 + \hat{\beta}\|s\|_2. \tag{5.15}$$

It is known [63] that s^* minimizes $z(s)$ if and only if **0** is a subgradient of the functional $z(s)$ at the point s^*, i.e.,

$$\mathbf{0} \in \partial z(s^*) = 2(s^* - \hat{s}) + \hat{\beta}\partial\|s^*\|_2, \tag{5.16}$$

where $\partial\|s^*\|_2$ denotes the subdifferential of $\|s\|_2$ at s^*. Moreover, we can verify [73] that

$$\partial\|s\|_2 = \left\{v \in \mathbb{R}^d : v = \frac{s}{\|s\|_2} \text{ if } s \neq 0; \|v\|_2 \leq 1 \text{ if } s = 0\right\}, \forall s \in \mathbb{R}^d.$$

If $s^* \neq 0$, it follows from Eq. (5.16) that

$$2\left(s^* - \hat{s}\right) + \hat{\beta}\frac{s^*}{\|s^*\|} = 0. \tag{5.17}$$

By rearranging Eq. (5.17) into the equality $s^*(2 + \hat{\beta}/\|s^*\|_2) = 2\hat{s}$ and taking the Euclidean norm for both sides, we have

$$\|s^*\|_2 = \|\hat{s}\|_2 - \frac{\hat{\beta}}{2},\ \|\hat{s}\|_2 > \frac{\hat{\beta}}{2}.$$

It follows that

$$s^* = \hat{s}\left(1 - \frac{\hat{\beta}}{2\|\hat{s}\|_2}\right),\ z(s^*) = \hat{\beta}\|\hat{s}\|_2 - \frac{\hat{\beta}^2}{4}. \tag{5.18}$$

If $s^* = 0$, we have

$$s^* = 0,\ z(s^*) = \|\hat{s}\|_2^2. \tag{5.19}$$

Since $z(s)$ is strictly convex with respect to the variable s, the problem in Eq. (5.14) admits a unique minimizer. From Eqs. (5.18) and (5.19), we have $\hat{\beta}\|\hat{s}\|_2 - \frac{\hat{\beta}^2}{4} - \|\hat{s}\|_2^2 \leq 0$. We complete the proof. □

The computation cost of solving Eq. (5.13) is relatively small compared to the cost of solving Eq. (5.12).

Main Algorithm

The pseudo-codes of the APM algorithm are presented in Algorithm 11. It is well known [45, 57, 71] that Algorithm 11 globally converges at the rate of $\mathcal{O}(\frac{1}{k^2})$, which is optimal among all first-order methods. In Algorithm 11, k denotes the iteration index, (L_k, S_k) denotes the feasible solution pair, $(\widetilde{L}_k, \widetilde{S}_k)$ denotes the searching point pair, and the stopping criterion can be set as: the change of the objective value of RMTL in two successive iterations is smaller than some pre-specified positive value ϵ.

```
Input: $L_0 \in \mathbb{R}^{d \times m}$, $S_0 \in \mathbb{R}^{d \times m}$, $\gamma_0 \in \mathbb{R}$, $\hat{k} \in \mathbb{R}$.
Output: $L_k \in \mathbb{R}^{d \times m}$, $S_k \in \mathbb{R}^{d \times m}$.
Set $L_1 = L_0$, $S_1 = S_0$, $t_{-1} = 0$, and $t_0 = 1$.
for $k = 1, 2, \cdots, \hat{k}$ do
    Compute $\alpha_k = \frac{t_{k-2} - 1}{t_{k-1}}$.
    Compute $(\widetilde{L}_k, \widetilde{S}_k)$ as
        $\widetilde{L}_k = (1 + \alpha_k) L_k - \alpha_k L_{k-1}$,
        $\widetilde{S}_k = (1 + \alpha_k) S_k - \alpha_i S_{k-1}$.
    while (true)
        Compute $(L_k, S_k)$ via Eqs. (6.7) and (5.11).
        if Eq. (5.10) is satisfied then exit the while loop
            else update $\gamma_k$ as $\gamma_k \leftarrow \gamma_k \times 2$.
        end-if
    end-while
    if stopping criterion satisfied then exit the loop.
    Update $t_k = \frac{1}{2}(1 + \sqrt{1 + 4t_{k-1}^2})$.
end-for
```

Algorithm 11: Accelerated Proximal Method for RMTL

5.4 Theoretical Analysis

In this section, we derive a performance bound for the proposed RMTL formulation in Eq. (5.4). This performance bound can be used to theoretically evaluate how well the integration of the low-rank structure and the group-sparse structure can estimate the multiple tasks (the ground truth of the linear predictive functions). Note that in the following analysis, for simplicity we assume that the training sample sizes for all tasks are the same; the derivation below can be easily extended to the setting where the training sample size for each task is different.

Assume that the linear predictive function associated with the i-th task satisfies

$$y_j^i = f_i(x_j^i) + \delta_{ij} = w_i^T x_j^i + \delta_{ij},\ i \in \mathbb{N}_m,\ j \in \mathbb{N}_n, \tag{5.20}$$

where $\{(x_j^i, y_j^i)\}$ are the training data pairs of the i-th task, and $\delta_{ij} \sim \mathcal{N}(0, \sigma_\delta^2)$ is a stochastic noise variable. For the i-th task, denote its training data matrix X_i and its label vector y_i respectively by

$$X_i = [x_1^i, \cdots, x_n^i] \in \mathbb{R}^{d \times n}, y_i = [y_1^i, \cdots, y_n^i]^T \in \mathbb{R}^n, i \in \mathbb{N}_m. \tag{5.21}$$

Denote the empirical evaluation of the i-th task f_i over the training data $\{x_j^i\}$ and the associated noise vector δ_i respectively by

$$\hat{f}_i = [f_i(x_1^i), \cdots, f_i(x_n^i)]^T \in \mathbb{R}^n,\ \ \delta_i = [\delta_{i1}, \cdots, \delta_{in}]^T \in \mathbb{R}^n. \tag{5.22}$$

It follows that Eq. (6.1) can be expressed in a compact form as

$$y_i = \hat{f}_i + \delta_i,\ i \in \mathbb{N}_m. \tag{5.23}$$

Moreover, the optimization problem in Eq. (5.4) can be rewritten as

$$(\widehat{L}_z, \widehat{S}_z) = \arg\min_{L,S} \frac{1}{mn} \sum_{i=1}^{m} \|X_i^T(l_i + s_i) - y_i\|_2^2 + \alpha\|L\|_* + \beta\|S\|_{1,2}, \tag{5.24}$$

where $\widehat{L}_z = [\hat{l}_1, \cdots, \hat{l}_m]$ and $\widehat{S}_z = [\hat{s}_1, \cdots, \hat{s}_m]$ are the optimal solution pair obtained via solving Eq. (5.24).

Basic Properties of the Optimal Solution

We present some basic properties of the optimal solution pair defined in Eq. (5.24); these properties are important building blocks for our following theoretical analysis. We first define two operators, namely $\mathcal{Q}$ and its complement $\mathcal{Q}_\perp$, on an arbitrary matrix pair (of the same size), based on Lemma 3.4 in [94].

Lemma 5.4.1. *Given any L and $\widehat{L}$ of the same size $d \times m$, let rank$(L) = r \leq \min(d, m)$ and denote the SVD of L as*

$$L = U \begin{bmatrix} \Sigma & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} V^T,$$

where $U \in \mathbb{R}^{d \times d}$ and $V \in \mathbb{R}^{m \times m}$ are orthogonal, and $\Sigma \in \mathbb{R}^{r \times r}$ is diagonal consisting of the non-zero singular values on its main diagonal. Let

$$U^T(\widehat{L} - L)V = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where $M_{11} \in \mathbb{R}^{r\times r}$, $M_{12} \in \mathbb{R}^{r\times(m-r)}$, $M_{21} \in \mathbb{R}^{(d-r)\times r}$, and $M_{22} \in \mathbb{R}^{(d-r)\times(m-r)}$. Define $\mathcal{Q}$ and $\mathcal{Q}_\perp$ on $\widehat{L} - L$ as

$$\mathcal{Q}(\widehat{L} - L) = U \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & \boldsymbol{0} \end{bmatrix} V^T, \mathcal{Q}_\perp(\widehat{L} - L) = U \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & M_{22} \end{bmatrix} V^T.$$

Then rank$(\mathcal{Q}(\widehat{L} - L)) \leq 2r$, $L\mathcal{Q}_\perp^T(\widehat{L} - L) = L^T\mathcal{Q}_\perp(\widehat{L} - L) = 0$.

The results in Lemma 6.4.1 imply a condition under which the trace norm on a matrix pair is additive. From Lemma 6.4.1 we can verify

$$\|L + \mathcal{Q}_\perp(\widehat{L} - L)\|_* = \|L\|_* + \|\mathcal{Q}_\perp(\widehat{L} - L)\|_*, \tag{5.25}$$

for arbitrary L and $\widehat{L}$ of the same size. As a direct consequence of Lemma 6.4.1, we derive a bound on the trace norm of the matrices of interest as summarized below.

Corollary 5.4.1. *For an arbitrary matrix pair $\widehat{L}$ and L, the following inequality holds*

$$\|\widehat{L} - L\|_* + \|L\|_* - \|\widehat{L}\|_* \leq 2\|\mathcal{Q}(\widehat{L} - L)\|_*.$$

Proof. From Lemma 6.4.1, we have

$$\widehat{L} - L = \mathcal{Q}(\widehat{L} - L) + \mathcal{Q}_\perp(\widehat{L} - L)$$

for any matrix pair L and $\widehat{L}$. It follows that

$$\begin{aligned} \|\widehat{L}\|_* &= \|L + \mathcal{Q}(\widehat{L} - L) + \mathcal{Q}_\perp(\widehat{L} - L)\|_* \\ &\geq \|L + \mathcal{Q}_\perp(\widehat{L} - L)\|_* - \|\mathcal{Q}(\widehat{L} - L)\|_* \\ &= \|L\|_* + \|\mathcal{Q}_\perp(\widehat{L} - L)\|_* - \|\mathcal{Q}(\widehat{L} - L)\|_*, \end{aligned}$$

where the inequality above follows from the triangle inequality and the last equality above follows from Eq. (6.33). Moreover,

$$\begin{aligned} & \|\widehat{L} - L\|_* + \|L\|_* - \|\widehat{L}\|_* \\ \leq\ & \|\widehat{L} - L\|_* + \|L\|_* - \Big(\|L\|_* + \|\mathcal{Q}_\perp(\widehat{L} - L)\|_* - \|\mathcal{Q}(\widehat{L} - L)\|_*\Big) \\ \leq\ & 2\|\mathcal{Q}(\widehat{L} - L)\|_*. \end{aligned}$$

We complete the proof of this corollary. □

Analogous to the bound on the trace norm derived in Corollary 6.4.1, we derive a bound on the $\ell_{1,2}$-norm of the matrices of interest. Denote by $\mathcal{C}(S)$ the set of indices corresponding to the non-zero columns of the matrix S as

$$\mathcal{C}(S) = \{i : s_i \neq 0,\, i \in \mathbb{N}_m\}, \tag{5.26}$$

and by $\mathcal{C}_\perp(S)$ the associated complement (the set of indices corresponding to the zero columns). Denote by $\widehat{S}_{\mathcal{C}(S)}$ the matrix of the same columns as $\widehat{S}$ on the index set $\mathcal{C}(S)$ and of zero columns on the index set $\mathcal{C}_\perp(S)$, i.e., $\widehat{S}_{\mathcal{C}(S)} = [\tilde{s}_1, \cdots, \tilde{s}_m]$, where $\tilde{s}_i = \hat{s}_i$ if $i \in \mathcal{C}(S)$ and $\tilde{s}_i = \mathbf{0}$ if $i \in \mathcal{C}_\perp(S)$. The bound on the $\ell_{1,2}$-norm is summarized below.

Lemma 5.4.2. *Given a matrix pair S and $\widehat{S}$ of the same size, the following inequality holds*

$$\|\widehat{S} - S\|_{1,2} + \|S\|_{1,2} - \|\widehat{S}\|_{1,2} \leq 2\|(\widehat{S} - S)_{\mathcal{C}(S)}\|_{1,2}. \tag{5.27}$$

Proof. From the definition of $\mathcal{C}(S)$ in Eq. (5.26), we have

$$S_{\mathcal{C}_\perp(S)} = \mathbf{0}, \;\; \|(\widehat{S} - S)_{\mathcal{C}_\perp(S)}\|_{1,2} = \|\widehat{S}_{\mathcal{C}_\perp(S)}\|_{1,2}.$$

It follows that

$$\begin{aligned} & \|(\widehat{S} - S)_{\mathcal{C}_\perp(S)}\|_{1,2} + \|S\|_{1,2} - \|\widehat{S}\|_{1,2} \\ = \;& \|\widehat{S}_{\mathcal{C}_\perp(S)}\|_{1,2} + \|S\|_{1,2} - \|\widehat{S}\|_{1,2} \\ = \;& \|S_{\mathcal{C}(S)}\|_{1,2} - \|\widehat{S}_{\mathcal{C}(S)}\|_{1,2} \\ \leq \;& \|(S - \widehat{S})_{\mathcal{C}(S)}\|_{1,2} \\ = \;& \|(\widehat{S} - S)_{\mathcal{C}(S)}\|_{1,2}. \end{aligned}$$

By substituting the equation above into the left side of Eq. (5.27), we complete the proof of this lemma. □

We now present some important properties of the optimal solution in Eq. (5.24) as summarized in the following lemma.

Lemma 5.4.3. *Consider the optimization problem in Eq. (5.24) for $m \geq 2$ and $n, d \geq 1$. Let X_i and y_i be defined in Eq. (5.21), and $\hat{f}_i$ and δ_i be defined in Eq. (5.22). Assume that all diagonal elements of the matrix $X_i X_i^T$ are equal to 1 (features are normalized). Take the regularization parameters α and β as*

$$\frac{\alpha}{\sqrt{m}}, \beta \geq \lambda, \;\; \lambda = \frac{2\sigma_\delta}{nm}\sqrt{d+t}, \tag{5.28}$$

where $t > 0$ is a universal constant. Then with probability of at least $1-m\exp\left(-\frac{1}{2}\left(t-d\log\left(1+\frac{t}{d}\right)\right)\right)$, for a global minimizer $\widehat{L}_z, \widehat{S}_z$ in Eq. (5.24) and any $L, S \in \mathbb{R}^{d\times m}$, we have

$$\frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(\hat{l}_i+\hat{s}_i)-\hat{f}_i\|_2^2 \leq \frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(l_i+s_i)-\hat{f}_i\|_2^2 + \alpha\|\mathcal{Q}(\widehat{L}_z-L)\| + \beta\|(\widehat{S}_z-S)_{\mathcal{C}(S)}\|_{1,2}, \quad (5.29)$$

where $\hat{l}_i$ and $\hat{s}_i$ (l_i and s_i) are the i-th columns of $\widehat{L}_z$ and $\widehat{S}_z$ (L and S), respectively.

Proof. From the definition of $(\widehat{L}_z, \widehat{S}_z)$ in Eq. (5.24), we have

$$\frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(\hat{l}_i+\hat{s}_i)-y_i\|_2^2 \leq \frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(l_i+s_i)-y_i\|_2^2$$
$$\alpha\|L\|_* + \beta\|S\|_{1,2} - \alpha\|\widehat{L}_z\|_* - \beta\|\widehat{S}_z\|_{1,2}.$$

By substituting Eq. (5.23) into the inequality above and rearranging all terms, we have

$$\frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(\hat{l}_i+\hat{s}_i)-\hat{f}_i\|_2^2 \leq \frac{1}{nm}\sum_{i=1}^{m}\|X_i^T(l_i+s_i)-\hat{f}_i\|_2^2+\alpha(\|L\|_*-\|\widehat{L}_z\|_*)+\beta(\|S\|_{1,2}-\|\widehat{S}_z\|_{1,2}) + \frac{2}{nm}\sum_{i=1}^{m}\langle\hat{l}_i-l_i, X_i\delta_i\rangle + \frac{2}{nm}\sum_{i=1}^{m}\langle\hat{s}_i-s_i, X_i\delta_i\rangle. \quad (5.30)$$

Next we compute upper bounds for the terms $\frac{2}{nm}\sum_{i=1}^{m}\langle\hat{l}_i-l_i, X_i\delta_i\rangle$ and $\frac{2}{nm}\sum_{i=1}^{m}\langle\hat{s}_i-s_i, X_i\delta_i\rangle$ in Eq. (5.30), respectively. Define a set of random events $\{\mathcal{A}_i\}$ as

$$\mathcal{A}_i = \left\{\frac{2}{nm}\|X_i\delta_i\|_2 \leq \lambda\right\},\ \forall i \in \mathbb{N}_m.$$

For each $\mathcal{A}_i$, define a set of random variables $\{v_{ij}\}$ as

$$v_{ij} = \frac{1}{\sigma_\delta}\sum_{k=1}^{n} x_{jk}^i\delta_{ik},\ j \in \mathbb{N}_d,$$

where x_{jk}^i denotes the (j,k)-th entry of the data matrix X_i. Since all diagonal elements of the matrix $X_iX_i^T$ are equal to 1, it can be shown that $\{v_{i1}, v_{i2}, \cdots, v_{id}\}$ are i.i.d. Gaussian variables obeying $\mathcal{N}(0,1)$ (Lemma 1 in the Appendix). We can also verify that $\sum_{j=1}^{d} v_{ij}^2$ is a chi-squared random variable with d degrees of freedom. Moreover taking λ as in Eq. (6.37), we have

$$\begin{aligned}\Pr\left(\frac{2}{nm}\|X_i\delta_i\|_2 > \lambda\right) &= \Pr\left(\sum_{j=1}^{d}\left(\sum_{k=1}^{n} x_{jk}^i\delta_{ik}\right)^2 \geq \frac{\lambda^2n^2m^2}{4}\right) \\ &= \Pr\left(\sum_{j=1}^{d} v_{ij}^2 \geq d+t\right) \leq \exp\left(-\frac{1}{2}\mu_d^2(t)\right),\end{aligned}$$

where $\mu_d(t) = \sqrt{t - d\log\left(1 + \frac{t}{d}\right)}$ $(t > 0)$, and the last inequality above follows from a concentration inequality (Lemma 6.4.3 in the Appendix). Let $\mathcal{A} = \bigcap_{i=1}^m \mathcal{A}_i$. Denote by $\mathcal{A}_i^c$ the complement of each event $\mathcal{A}_i$. It follows that

$$\Pr(\mathcal{A}) \geq 1 - \Pr\left(\bigcup_{i=1}^m \mathcal{A}_i^c\right) \geq 1 - m\exp\left(-\frac{1}{2}\mu_d^2(t)\right).$$

Under the event $\mathcal{A}$, we derive a bound on the term $\frac{2}{nm}\sum_{i=1}^m \langle \hat{l}_i - l_i, X_i\delta_i\rangle$ as

$$\begin{aligned} \frac{2}{nm}\sum_{i=1}^m \langle \hat{l}_i - l_i, X_i\delta_i\rangle &\leq \frac{2}{nm}\sum_{i=1}^m \|\hat{l}_i - l_i\|_2 \|X_i\delta_i\|_2 \\ &\leq \lambda\sum_{i=1}^m \|\hat{l}_i - l_i\|_2 \leq \alpha\|\widehat{L}_z - L\|_*, \end{aligned} \tag{5.31}$$

where the first inequality above follows from Cauchy-Schwarz inequality and the second inequality follows from

$$\begin{aligned} \sum_{i=1}^m \|\hat{l}_i - l_i\|_2 &\leq \sqrt{m\sum_{i=1}^m \|\hat{l}_i - l_i\|_2^2} \\ &= \sqrt{m}\|\widehat{L}_z - L\|_F \leq \sqrt{m}\|\widehat{L}_z - L\|_*. \end{aligned}$$

Similarly under $\mathcal{A}$, we also derive a bound on the term $\frac{2}{nm}\sum_{i=1}^m \langle \hat{s}_i - s_i, X_i f_i\rangle$ as

$$\begin{aligned} \frac{2}{nm}\sum_{i=1}^m \langle \hat{s}_i - s_i, X_i\delta_i\rangle &\leq \frac{2}{nm}\sum_{i=1}^m \|\hat{s}_i - s_i\|_2 \|X_i\delta_i\|_2 \\ &\leq \beta\|\widehat{S}_z - S\|_{1,2}. \end{aligned} \tag{5.32}$$

Moreover we bound the right side of Eq.(5.30) using the results from Eqs. (5.31) and (5.32). It follows that

$$\frac{1}{nm}\sum_{i=1}^m \|X_i^T(\hat{l}_i + \hat{s}_i) - \hat{f}_i\|_2^2 \leq \frac{1}{nm}\sum_{i=1}^m \|X_i^T(l_i + s_i) - \hat{f}_i\|_2^2 +$$
$$\alpha(\|\widehat{L}_z - L\|_* + \|L\|_* - \|\widehat{L}_z\|_*) + \beta(\|\widehat{S}_z - S\|_{1,2} + \|S\|_{1,2} - \|\widehat{S}_z\|_{1,2}).$$

Finally by applying Corollary 6.4.1 and Lemma 5.4.2 together with the inequality above, we complete the proof. □

Performance Bound

We present a performance bound of the proposed RMTL formulation in Eq. (5.24). This bound measures how well the multi-task learning scheme (via the integration of the low-rank structure and the $\ell_{1,2}$-norm structure) can estimate the linear predictive functions in Eq. (6.1).

We begin with some notations. Let $X \in \mathbb{R}^{md\times mn}$ be a block-diagonal matrix with its i-th block formed by the matrix $X_i \in \mathbb{R}^{d\times n}$ $(i \in \mathbb{N}_m)$. Define a diagonalization operator $\mathcal{D}$ on an arbitrary $\Omega = [\omega_1, \omega_2, \cdots, \omega_m] \in \mathbb{R}^{d\times m}$: $\mathcal{D}(\Omega) \in \mathbb{R}^{md\times m}$ is a block diagonal matrix with its i-th block formed by the column vector $\omega_i \in \mathbb{R}^d$. Let $\mathcal{F} = [\hat{f}_1, \cdots, \hat{f}_m]$, where $\hat{f}_i$ is defined in Eq. (5.22). Therefore we can rewrite Eq. (5.29) in a compact form as

$$\frac{1}{T}\|X^T\mathcal{D}(\widehat{L}_z + \widehat{S}_z) - \mathcal{D}(\mathcal{F})\|_F^2 \leq \frac{1}{T}\|X^T\mathcal{D}(L+S) - \mathcal{D}(\mathcal{F})\|_F^2 + \alpha\|\mathcal{Q}(\widehat{L}_z - L)\|_* + \beta\|(\widehat{S}_z - S)_{\mathcal{C}(S)}\|_{1,2}, \quad (5.33)$$

where $T = nm$. We next introduce our assumption over a restricted set. The assumption is associated with training data X and the geometric structure of the matrices of interest.

Assumption 5.4.1. *For a matrix pair Γ_L and Γ_S of size d by m, let $s \leq \min(d,m)$ and $q \leq m$. We assume that there exist constants $\kappa_1(s)$ and $\kappa_2(q)$ such that*

$$\kappa_1(s) \triangleq \min_{\Gamma_L,\Gamma_S \in \mathcal{R}(s,q)} \frac{\|X\mathcal{D}(\Gamma_L + \Gamma_S)\|_F}{\sqrt{T}\|\mathcal{Q}(\Gamma_L)\|_*} > 0, \quad (5.34)$$

$$\kappa_2(q) \triangleq \min_{\Gamma_L,\Gamma_S \in \mathcal{R}(s,q)} \frac{\|X\mathcal{D}(\Gamma_L + \Gamma_S)\|_F}{\sqrt{T}\|(\Gamma_S)_{\mathcal{C}(S)}\|_{1,2}} > 0, \quad (5.35)$$

where the restricted set $\mathcal{R}(s,q)$ is defined as

$$\mathcal{R}(s,q) = \left\{\Gamma_L, \Gamma_S \in \mathbb{R}^{d\times m} \,|\, \Gamma_L \neq 0,\, \Gamma_S \neq 0, \textit{rank}(\mathcal{Q}(\Gamma_L)) \leq s,\ |\mathcal{C}(\Gamma_S)| \leq q\right\},$$

and $\mathcal{C}(\cdot)$ is defined in Eq. (5.26), and $|\widehat{C}|$ denotes the number of elements in the set $\widehat{C}$.

The assumption in Eqs. (6.40) and (5.35) can be implied by several sufficient conditions as in [95]. Due to the space constraint, the details are omitted. Note that similar assumptions are used in [96] for deriving a certain performance bound for a different multi-task learning formulation.

We present the performance bound of the RMTL formulation in the following theorem.

Theorem 5.4.1. *Consider the optimization problem in Eq. (5.24) for $m \geq 2$ and $n, d \geq 1$. Take the regularization parameters α and β as in Eq. (6.37). Then with probability of at least $1 - m\exp\left(-\frac{1}{2}\left(t - d\log\left(1 + \frac{t}{d}\right)\right)\right)$, for a global minimizer $\widehat{L}_z, \widehat{S}_z$ in Eq. (5.24), we have*

$$\frac{1}{T}\|X\mathcal{D}(\widehat{L}_z + \widehat{S}_z) - \mathcal{D}(\mathcal{F})\|_F^2 \leq (1+\epsilon)\inf_{L,S}\frac{1}{T}\|X\mathcal{D}(L+S) - \mathcal{D}(\mathcal{F})\|_F^2 + \mathcal{E}(\epsilon)\left(\frac{\alpha^2}{\kappa_1^2(2r)} + \frac{\beta^2}{\kappa_2^2(c)}\right), \quad (5.36)$$

where $\inf$ is taken over all $L, S \in \mathbb{R}^{d\times m}$ with rank$(L) \leq r$ and $|\mathcal{C}(S)| \leq c$, and $\mathcal{E}(\epsilon) > 0$ is a constant depending only on ϵ.

Proof. Denote $\Gamma_L = \widehat{L}_z - L$ and $\Gamma_S = \widehat{S}_z - S$. It follows from Eq. (5.33) that

$$\frac{1}{T}\|X^T\mathcal{D}(\widehat{L}_z+\widehat{S}_z)-\mathcal{D}(\mathcal{F})\|_F^2 \leq \frac{1}{T}\|X^T\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F^2+\alpha\|\mathcal{Q}(\Gamma_L)\|_*+\beta\|(\Gamma_S)_{\mathcal{C}(S)}\|_{1,2}. \quad (5.37)$$

Given $\mathcal{Q}(\Gamma_L) \leq 2r$ (from Lemma 6.4.1) and $|\mathcal{C}(S)| \leq c$, we derive upper bounds on $\alpha\|\mathcal{Q}(\Gamma_L)\|_*$ and $\beta\|(\Gamma_S)_{\mathcal{C}(S)})\|_{1,2}$ over the restrict set $\mathcal{R}(2r, c)$ based on Assumptions 6.4.1, respectively. It follows from Eq. (6.40) in Assumption 6.4.1 that

$$\begin{aligned}
2\alpha\|\mathcal{Q}(\Gamma_L)\|_* &\leq \frac{2\alpha}{\kappa_1(2r)\sqrt{T}}\|X\mathcal{D}(\Gamma_L+\Gamma_S)\|_F \\
&\leq \frac{2\alpha}{\kappa_1(2r)\sqrt{T}}\left(\|X\mathcal{D}(\widehat{L}_z+\widehat{S}_z)-\mathcal{D}(\mathcal{F})\|_F+\|X\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F\right) \\
&\leq \frac{\alpha^2\tau}{\kappa_1^2(2r)}+\frac{1}{\tau T}\|X\mathcal{D}(\widehat{L}_z+\widehat{S}_z)-\mathcal{D}(\mathcal{F})\|_F^2+\frac{\alpha^2\tau}{\kappa_1^2(2r)}+ \\
&\quad \frac{1}{\tau T}\|X\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F^2, \qquad (5.38)
\end{aligned}$$

where the last inequality above follows from $2ab \leq a^2\tau + b^2\frac{1}{\tau}$ for $\tau > 0$. Similarly, we have

$$\begin{aligned}
2\beta\|(\Gamma_S)_{\mathcal{C}(S)}\|_{1,2} &\leq \frac{\beta^2\tau}{\kappa_2^2(c)}+\frac{1}{\tau T}\|X\mathcal{D}(\widehat{L}_z+\widehat{S}_z)-\mathcal{D}(\mathcal{F})\|_F^2+ \\
&\quad \frac{\beta^2\tau}{\kappa_2^2(c)}+\frac{1}{\tau T}\|X\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F^2. \qquad (5.39)
\end{aligned}$$

Substituting Eqs. (6.43) and (6.44) into Eq. (5.37) and setting $\tau = 2 + \frac{4}{\epsilon}$, we obtain

$$\begin{aligned}
&\frac{1}{T}\|X\mathcal{D}(\widehat{L}_z+\widehat{S}_z)-\mathcal{D}(\mathcal{F})\|_F^2 \\
\leq\ & \frac{\tau+2}{\tau-2}\|X\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F^2+\frac{2\tau^2}{\tau-2}\left(\frac{\alpha^2}{\kappa_1^2(2r)}+\frac{\beta^2}{\kappa_2^2(c)}\right) \\
=\ & (1+\epsilon)\|X\mathcal{D}(L+S)-\mathcal{D}(\mathcal{F})\|_F^2+\mathcal{E}(\epsilon)\left(\frac{\alpha^2}{\kappa_1^2(2r)}+\frac{\beta^2}{\kappa_2^2(c)}\right),
\end{aligned}$$

where $\mathcal{E}(\epsilon) = \epsilon(\frac{1}{2}+\frac{1}{\epsilon})^2$. This completes the proof. □

The performance bound described in Eq. (6.41) can be refined by choosing specific values for the regularization parameters α and β: it can be verified that the component $\frac{\alpha^2}{\kappa_1^2(2r)} + \frac{\beta^2}{\kappa_2^2(c)}$ is minimized if α and β are chosen to be proportional to $\kappa_1^2(2r)$ and $\kappa_2^2(c)$, respectively.

5.5 Experiments

In this section, we evaluate the proposed RMTL formulation in Eq. (5.4) in comparison with other representative algorithms for multi-task learning; we also conduct numerical studies on the APM algorithm in comparison with the commonly used proximal method (PM) [45, 57] for solving RMTL. All algorithms are implemented in Matlab. Note that for numerical accuracy consideration, we solve the RMLT formulation with its objective function multiplied by nm, where m and n correspond to the task number and the sum of the sample sizes for all tasks, respectively.

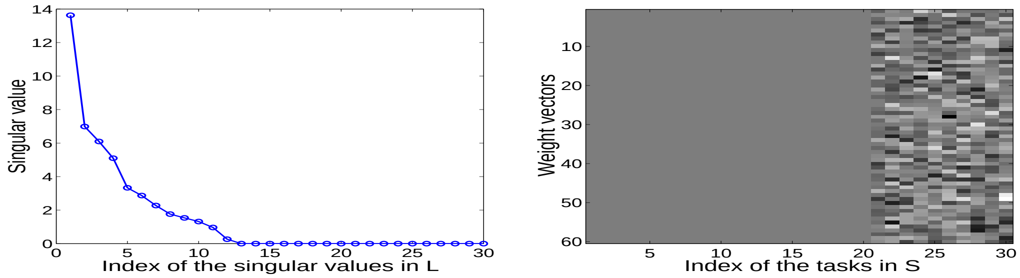


Figure 5.1: Demonstration of the extracted low-rank and group structures: the left plot shows the singular values of the low-rank component L (the last 18 singular values are zero); the right plot demonstrates the structure of the group-sparse component S (the first 20 columns are zero-vectors). In the right plot the grey area corresponds to the pixels of zero-value.

Demonstration of Extracted Structures

We apply the RMTL algorithm on a synthetic data set and then demonstrate the extracted low-rank and group-sparse structures. The synthetic data is constructed as follows: set the task number $m = 30$, the size of the training samples for each task $n_i = 50$, and the feature dimensionality of the training samples $d = 60$; generate the entries of the training data $X_i \in \mathbb{R}^{d \times n_i}$ (for the i-th task) randomly from the distribution $\mathcal{N}(0, 25)$; generate the entries in the low-rank component L (of size $d \times m$) randomly from $\mathcal{N}(0, 16)$ and then set its smallest 20 singular values at 0; generate the entries in the group-sparse component S (of size $d \times m$) randomly from $\mathcal{N}(0, 20)$ and then set its first 20 columns as zero-vectors; construct the response (target) vector of each task as $y_i = X_i^T(L + S) + \delta_i \in \mathbb{R}^{n_i}$ $(i \in \mathbb{N}_m)$, where each entry in the vector δ_i is randomly generated from $\mathcal{N}(0, 1)$. Under this experimental setting, we construct 20 related tasks as well as 10 outlier tasks, where each task is associated with 50 training samples of feature dimensionality 60.

In Figure 5.1, we present the low-rank component L and the group-sparse component S obtained by solving RMTL with $\alpha = 50$ and $\beta = 10$. From the left plot of Figure 5.1, we can observe that the matrix L (of size 60×50) has 12 non-zero singular values; this result is consistent with our problem setting of using a low-rank structure to capture the tasks relationship. From the right plot of Figure 5.1, we can observe that the first 20 columns (corresponding to the related tasks) in S are zero vectors, while the last 10 columns (corresponding to the outlier tasks) are non-zero vectors. The results in Figure 5.1 empirically demonstrate the effectiveness of RMTL.

Table 5.1: Performance comparison of the six competing algorithms in terms of the normalized MSE (nMSE) and the averaged MSE (aMSE) with standard deviation using the School data. All parameters of the six methods are determined via cross-validation and the reported regression performance is averaged over 15 random repetitions. Note that a smaller value of nMSE and aMSE represents better regression performance.

Measure	training ratio	Ridge	Lasso	TraceNorm	Sparse-LowRank	CMTL	Robust MTL
nMSE	10%	1.039 ± 0.004	1.026 ± 0.013	0.936 ± 0.037	0.918 ± 0.026	0.941 ± 0.002	0.913 ± 0.004
	20%	0.877 ± 0.004	0.875 ± 0.019	0.821 ± 0.003	0.813 ± 0.013	0.833 ± 0.004	0.806 ± 0.010
	30%	0.817 ± 0.009	0.814 ± 0.009	0.787 ± 0.001	0.766 ± 0.009	0.792 ± 0.005	0.760 ± 0.003
aMSE	10%	0.271 ± 0.002	0.268 ± 0.004	0.250 ± 0.010	0.242 ± 0.008	0.255 ± 0.003	0.233 ± 0.002
	20%	0.230 ± 0.000	0.229 ± 0.005	0.216 ± 0.002	0.211 ± 0.004	0.213 ± 0.007	0.202 ± 0.003
	30%	0.216 ± 0.002	0.214 ± 0.001	0.209 ± 0.001	0.201 ± 0.002	0.192 ± 0.010	0.182 ± 0.001

Performance Evaluation of RMTL

We evaluate the RMTL algorithm on multi-task regression problems in comparison with other representative algorithms including ridge regression (Ridge), least squares with ℓ_1-norm regularization (Lasso), least squares with trace norm regularization (TraceNorm), least squares with low-rank and sparse structures regularization (Sparse-LowRank) [88], and convex multi-task feature learning (CMTL) [53]. The normalized mean squared error (nMSE) and the averaged mean squared error (aMSE) are employed as the regression performance measures as used in previous studies [53, 81]. Note that nMSE is defined as the mean squared error (MSE) divided by the variance of the target vector; aMSE is defined as MSE divided by the squared norm of the target vector. We adopt APM to solve RMTL and terminate APM when the relative change of the objective values in two successive iterations is smaller than 10^{-5}. We use the School data[1] and the SARCOS data[2] for the experiments.

The School data consists of the exam scores of 15362 students from 139 secondary schools; each student is described by 27 attributes such as gender and ethnic group. The exam score prediction of the students can be cast into a multi-task regression (learning) problem: we are given 139 tasks (schools), where each task has a different number of samples (students) and each sample has 27 features (attributes). We randomly select 10%, 20%, and 30% of the samples (from each task) to form the training set and use the rest of the samples as the test set. The experimental results averaged over 15 random repetitions are presented in Table 5.1. From the presented results, we have the following observations: (1) RMTL outperforms all other competing algorithms in terms of nMSE and aMSE; (2) the multi-task learning algorithms (TraceNorm, Sparse-LowRank, CMTL, and RMTL) outperform the single-task learning algorithms (Ridge and Lasso) in terms of both nMSE and

[1] `http://www.cs.ucl.ac.uk/staff/A.Argyriou/code/`

[2] `http://www.gaussianprocess.org/gpml/data/`

Table 5.2: Performance comparison of the six competing algorithms in terms of nMSE and aMSE with standard deviation using the SARCOS data.The experimental setting is similar to the one described in Table 5.1.

Measure	training size	Ridge	Lasso	TraceNorm	Sparse-LowRank	CMTL	Robust MTL
nMSE	50	0.245 ± 0.026	0.234 ± 0.018	0.226 ± 0.007	0.213 ± 0.003	0.219 ± 0.002	0.212 ± 0.004
	100	0.182 ± 0.014	0.162 ± 0.003	0.153 ± 0.002	0.149 ± 0.002	0.157 ± 0.004	0.146 ± 0.014
	150	0.150 ± 0.005	0.147 ± 0.003	0.132 ± 0.005	0.124 ± 0.001	0.130 ± 0.003	0.125 ± 0.002
aMSE	50	0.133 ± 0.014	0.123 ± 0.008	0.112 ± 0.006	0.107 ± 0.003	0.116 ± 0.001	0.098 ± 0.003
	100	0.105 ± 0.009	0.091 ± 0.002	0.081 ± 0.003	0.079 ± 0.005	0.085 ± 0.001	0.074 ± 0.008
	150	0.085 ± 0.005	0.082 ± 0.001	0.077 ± 0.002	0.066 ± 0.006	0.076 ± 0.003	0.067 ± 0.001

aMSE; (3) the performance of CMTL is similar to that of TraceNorm; this result may be due to the use of similar penalty terms in CMTL and TraceNorm.

The SARCOS data is collected for an inverse dynamics prediction problem for a seven degrees-of-freedom anthropomorphic robot arm. This data consists of 48933 observations corresponding to 7 joint torques; each of the observations is described by 21 features including 7 joint positions, 7 joint velocities, and 7 joint accelerations. Our goal is to construct mappings from each observation to 7 joint torques. We randomly select $50, 100, 150$ observations to form 3 training sets and accordingly randomly select 5000 observations to form 3 test sets. The experimental results averaged over 15 random repetitions are presented in Table 5.2. From the experimental results, we have the following observations: (1) RMTL performs better than or compares competitively to all other competing algorithms in terms of both nMSE and aMSE; (2) the multi-task learning algorithms (TraceNorm, Sparse-LowRank, CMTL, and RMTL) outperform the single-task learning algorithms (Ridge and Lasso) in terms of both nMSE and aMSE. We also observe that Sparse-LowRank has a similar performance to RMTL. In Sparse-LowRank, incoherent low-rank and (ℓ_1-norm based) sparse structures [88] are used to capture the task relatedness as well as identify discriminative features for each task. These results imply that allowing each task to independently select discriminative features may improve the robustness of the algorithm.

Sensitivity Studies on RMTL

We conduct a sensitivity study on the proposed RMTL formulation. In particular, we study how the regularization parameters and the training sample size affect the regression performance of RMTL in terms of nMSE and aMSE, respectively.

Effect of the Regularization Parameters For this experiment, we randomly select 10% of the School data as the training set and use the rest of the data as the test set. By fixing $\beta = 100$ as well as varying the value of α in α-value set, i.e., $[50 : 50 : 500]$, we study how the parameter α affects the regression performance of RMTL. Similarly, by fixing $\alpha = 150$ as well as varying the value of β in β-value set of $[20 : 5 : 115]$, we study how the parameter β affects the regression performance of RMTL. In Figure 5.2, we present the regression performance (averaged over 15 random repetitions) of RMTL in terms of nMSE (1st and 3rd plots) and aMSE (2nd and 4th plots) for each pair of (α, β). From Figure 5.2, we can observe that both nMSE and aMSE change with different settings of (α, β); we can also observe that the best performance of RMTL for a fixed α (or a fixed β) is obtained by setting β in the middle of β-value set (or setting the value of α in the middle of α-value set).

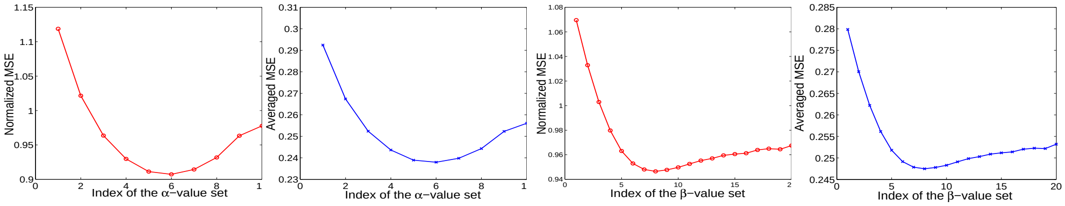


Figure 5.2: Sensitivity study on RMTL: study the effect of the parameters α and β in terms of nMSE (1st and 3rd plots) and aMSE (2nd and 4th plots), respectively. For the first two plots, we set $\beta = 100$ and vary α in the α-value set $[50 : 50 : 500]$; for the last two plots, we set $\beta = 150$ and vary β in the β-value set $[20 : 5 : 115]$.

Effect of the Training Ratio For this experiment, we randomly select $\{10\%, 20\%, \cdots, 80\%\}$ of the School data as the training set and use the rest of the data as the test set. We study how the the training sample size (in terms of the training ratio) affects the regression performance of RMTL. Note that the regularization parameters α and β are determined via double cross-validation. The experimental results are presented in Figure 5.3. We can observe that by increasing the training ratio, both the nMSE and aMSE decrease; this result is consist with our expectation that more training data will lead to more accurate predictive model and hence better generalization performance.

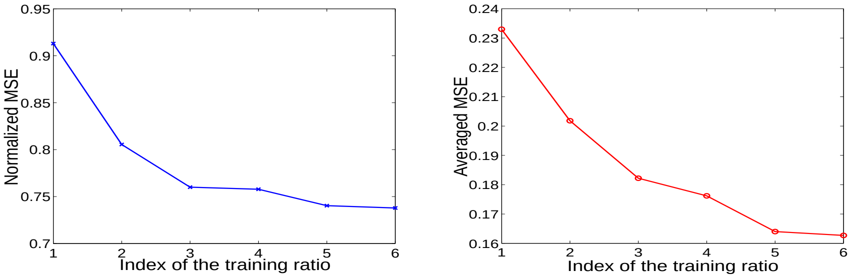


Figure 5.3: Sensitivity study on RMTL: study the effect of the training ratio in terms of nMSE (left plot) and aMSE (right plot), respectively. The regularization parameters are determined via double cross-validation. The i-th coordinate on the x-axis corresponds to the training ratio $i \times 10\%$.

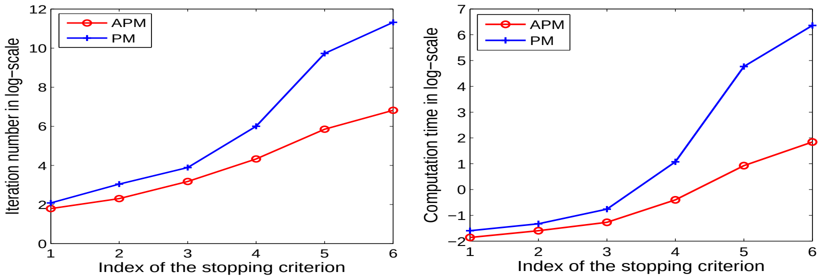


Figure 5.4: Computation cost comparison of APM and PM in terms of the iteration number (left plot) and the computation time in seconds (right plot) for solving RMTL. The i-th coordinate on the x-axis corresponds to the stopping criterion 10^{-i}.

We conduct numerical studies on APM in comparison with PM for solving RMTL in terms of the computation time (in seconds) and the iteration number. We randomly select 10% of the School data for the following experiments. The experimental setting is described as follows: we stop PM when the change of the objective value in two successive iterations is smaller than 10^{-i} and record the attained objective value; such a value is then used as the stopping criterion in APM, that is, we stop APM when the attained objective value in APM is equal to or smaller than the one previously obtained from PM; we vary the stopping criterion of PM in the set $\{10^{-i}\}_{i=1}^{6}$ and record the required computation time and iteration number for both PM and APM. From the experimental results presented in Figure 5.4, we have the following observations: (1) APM requires less computation time and iteration number than PM for attaining the same objective value; (2) both APM and PM require more computation time and a larger iteration number if the stopping criterion is set as a smaller value (higher accuracy).

5.6 Summary

In this chapter, we propose a robust multi-task learning (RMTL) algorithm which learns multiple tasks simultaneously as well as identifies the outlier tasks. The proposed RMTL algorithm captures the task relationships using a low-rank structure, and simultaneously identifies the outlier tasks using a group-sparse structure. RMTL is formulated as a non-smooth convex (unconstrained) optimization problem in which the least square loss is regularized by a combination of the trace norm regularization and the $\ell_{1,2}$-norm regularization. We propose to adopt the accelerated proximal method (APM) for solving this optimization problem and develop efficient algorithms for computing the associated proximal operator. We also conduct a theoretical analysis on the proposed RMTL formulation. In particular, we derive a key property of the optimal solution to RMTL; based on the key property, we establish a theoretical performance bound to characterize the learning performance of RMTL. Our experimental results on benchmark data sets demonstrate the effectiveness and efficiency of the proposed algorithms.

Appendix

Lemma 1. *Let $\delta_1, \delta_2, \cdots, \delta_n$ be a random sample of size n from the Gaussian distribution $\mathcal{N}(0, \sigma)$. Let $x_1, x_2, \cdots, x_n$ satisfy $x_1^2 + x_2^2 + \cdots + x_n^2 = 1$. Denote a random variable v as*

$$v = \frac{1}{\sigma} \sum_{i=1}^{n} x_i \delta_i.$$

Then v obeys the Gaussian distribution $\mathcal{N}(0, 1)$.

Proof. Since $\{\delta_i\}$ are mutually independent, the mean of the random variable v can be computed as

$$\mathbb{E}(v) = \mathbb{E}\left(\frac{1}{\sigma}\sum_{i=1}^{n} x_i \delta_i\right) = \frac{1}{\sigma}\sum_{i=1}^{n} x_i \mathbb{E}\left(\delta_i\right) = 0.$$

Similarly, the variance of v can be computed

$$\mathbb{E}\left(v - \mathbb{E}(v)\right)^2 = \mathbb{E}\left(\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i^2 \delta_i^2\right) = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i^2 \mathbb{E}\left(\delta_i^2\right) = 1,$$

where the first equality follows from $\mathbb{E}\left(\delta_i \delta_j\right) = 0 \ (i \neq j)$. Using the fact that the sum of Gaussian random variables is Gaussian distributed, we complete the proof of this lemma. □

Lemma 2. *Let $\mathcal{X}_p^2$ be a chi-squared random variable with p degrees of freedom. Then*

$$\Pr\left(\mathcal{X}_p^2 \geq p + \pi\right) \leq \exp\left(-\frac{1}{2}\left(\pi - p\log\left(1 + \frac{\pi}{p}\right)\right)\right), \pi > 0.$$

Proof. From Theorem 4.1 in [97], we approximate the chi-square distribution using a normal distribution as

$$\Pr\left(\mathcal{X}_p^2 \geq q\right) \leq \Pr\left(\mathcal{N}_{0,1} \geq z_p(q)\right), \, q > p,$$

where $\mathcal{N}_{0,1} \sim \mathcal{N}(0,1)$ and $z_p(q) = \sqrt{q - p - p\log\left(\frac{q}{p}\right)}$. It is known that for $x \sim \mathcal{N}(0,1)$, the inequality $\Pr\left(x \geq t\right) \leq \exp(-\frac{t^2}{2})$ holds. Therefore we have

$$\Pr\left(\mathcal{X}_p^2 \geq q\right) \leq \exp\left(-\frac{1}{2} z_p^2(q)\right).$$

By substituting $q = p + \pi \ (\pi > 0)$ into the inequality above, we complete the proof of this lemma. □

Chapter 6

Learning Multiple Tasks via Sparse Trace Norm Regularization

6.1 Introduction

We study the problem of estimating multiple predictive functions from noisy observations. Such a problem has received broad attention in many areas of statistics and machine learning [96, 98–100]. This line of work can be roughly divided into two categories: parametric estimation and nonparametric estimation; a common and important theme for both categories is the appropriate assumption of the structure in the model parameters (parametric setting) or the coefficients of the dictionary (nonparametric setting).

There has been an enormous amount of literature on effective function estimation based on different sparsity constraints, including the estimation of the sparse linear regression via ℓ_1-norm penalty [95, 98, 101, 102], and the estimation of the linear regression functions using group lasso estimator [96, 99]. More recently, trace norm regularization has become a popular tool for approximating a set of linear models and the associated low-rank matrices in the high-dimensional setting [100, 103]; the trace norm is the tightest convex surrogate [66] for the (non-convex) rank function under certain conditions, encouraging the sparsity in the singular values of the matrix of interest. One limitation of the use of trace norm regularization is that the resulting model is dense in general. However, in many real-world applications [104], the underlying structure of multiple predictive functions may be sparse as well as low-rank; the sparsity leads to explicitly interpretable prediction models and the low-rank implies essential subspace structure information. Similarly, the ℓ_1-norm is the tightest convex surrogate for the non-convex cardinality function [60], encouraging the sparsity in the entries of the matrix. This motivates us to explore the use of the combination of the trace norm and the ℓ_1-norm as a composite regularization (called sparse trace norm regularization) to induce the desirable sparse low-rank structure.

Trace norm regularization (minimization) has been investigated extensively in recent years. Efficient algorithms have been developed for solving convex programs with trace norm regularization [66, 105]; sufficient conditions for exact recovery from trace norm minimization have been established in [94]; consistency of trace norm minimization has been studied in [106]; trace norm minimization has been applied for matrix completion [107] and collaborative filtering [108, 109]. Similarly, ℓ_1-norm regularization has been well studied in the literature, just to mention a few, from the efficient algorithms for convex optimization [105, 110, 111], theoretical guarantee of the performance [102, 112], and model selection consistency [113].

In this chapter, we focus on estimating multiple predictive functions simultaneously from a finite dictionary of basis functions in the nonparametric regression setting. Our function estimation scheme assumes that each predictive function can be approximated using a linear combination of those basis functions. By assuming that the coefficient matrix of the basis functions admits a sparse low-rank structure, we formulate the function estimation problem as a convex formulation, in which the combination of the trace norm and the ℓ_1-norm is employed as a composite regularization to induce a sparse low-rank structure in the coefficient matrix. The simultaneous sparse and low-rank structure is different from the incoherent sparse and low-rank structures studied in [69, 114]. We propose to solve the function estimation problem using the accelerated gradient method and the alternating direction method of multipliers; we also develop efficient algorithms to solve the key components involved in both methods. We conduct theoretical analysis on the proposed convex formulation: we first present some basic properties of the optimal solution to the convex formulation (Lemma 6.4.4); we then present an assumption associated with the geometric nature of the basis functions over the prescribed observations; based on such an assumption, we derive a performance bound for the combined regularization for function estimation (Theorem 6.4.1). We conduct simulations on benchmark data to demonstrate the effectiveness and efficiency of the proposed algorithms.

6.2 Problem Formulation

Let $\{(x_1, y_1), \cdots, (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}^k$ be a set of prescribed sample pairs (fixed design) associated with k unknown functions $\{f_1, \cdots, f_k\}$ as

$$y_{ij} = f_j(x_i) + w_{ij}, \quad i \in \mathbb{N}_n,\, j \in \mathbb{N}_k, \tag{6.1}$$

where $f_j : \mathbb{R}^d \to \mathbb{R}$ is an unknown regression function, y_{ij} denotes the j-th entry of the response vector $y_i \in \mathbb{R}^k$, and $w_{ij} \sim \mathcal{N}(0, \sigma_w^2)$ is a stochastic noise variable. Let $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n \times d}$, $Y = [y_1, \cdots, y_n]^T \in \mathbb{R}^{n \times k}$, and $W = (w_{ij})_{i,j} \in \mathbb{R}^{n \times k}$. Denoting

$$\mathcal{F} = (f_j(x_i))_{i,j} \in \mathbb{R}^{n \times k}, \quad i \in \mathbb{N}_n,\, j \in \mathbb{N}_k, \tag{6.2}$$

we can rewrite Eq. (6.1) in a compact form as $Y = \mathcal{F} + W$. Let $\{g_1, \cdots, g_h\}$ be a set of h prespecified basis functions as $g_i : \mathbb{R}^d \to \mathbb{R}$, and let $\Theta = [\theta_1, \cdots, \theta_k] \in \mathbb{R}^{h \times k}$ be the coefficient matrix. We define

$$\hat{g}_j(x) = \sum_{i=1}^{h} \theta_{ij} g_i(x), \quad j \in \mathbb{N}_k, \tag{6.3}$$

where θ_{ij} denotes the i-th entry in the vector θ_j. Note that in practice the basis functions $\{g_i\}$ can be estimators from different methods, or different values of the tuning parameters of the same method.

We consider the problem of estimating the unknown functions $\{f_1, \cdots, f_k\}$ using the composite functions $\{\hat{g}_1, \cdots, \hat{g}_k\}$ defined in Eq. (6.3), respectively. Denote

$$\mathcal{G}_X = (g_j(x_i))_{i,j} \in \mathbb{R}^{n \times h}, \quad i \in \mathbb{N}_n,\, j \in \mathbb{N}_h, \tag{6.4}$$

and define the empirical error as

$$\widehat{S}(\Theta) = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} (\hat{g}_j(x_i) - y_{ij})^2 = \frac{1}{N} \|\mathcal{G}_X \Theta - Y\|_F^2, \tag{6.5}$$

where $N = n \times k$. Our goal is to estimate the model parameter Θ of a sparse low-rank structure from the given n sample pairs $\{(x_i, y_i)\}_{i=1}^n$. Such a structure induces the sparsity and the low rank simultaneously in a single matrix of interest.

Given that the functions $\{f_1, \cdots, f_k\}$ are coupled via Θ in some coherent sparse and low-rank structure, we propose to estimate Θ as

$$\widehat{\Theta} = \arg\min_{\Theta} \left(\widehat{S}(\Theta) + \alpha \|\Theta\|_* + \beta \|\Theta\|_1 \right), \tag{6.6}$$

where α and β are regularization parameters (estimated via cross-validation), and the linear combination of $\|\Theta\|_*$ and $\|\Theta\|_1$ is used to induce the sparse low-rank structure in Θ. The optimization problem in Eq. (6.6) is non-smooth convex and hence admits a globally optimal solution; it can be solved using many sophisticated optimization techniques [42, 66]; in Section 6.3, we propose to apply the accelerated gradient method [45] and the alternating direction method of multipliers [115] to solve the optimization problem in Eq. (6.6).

6.3 Optimization Algorithms

In this section, we consider to apply the accelerated gradient (AG) algorithm [45, 71, 116] and the alternating direction method of multipliers (ADMM) [115], respectively, to solve the (non-smooth and convex) optimization problem in Eq. (6.6). We also develop efficient algorithms to solve the key components involved in both AG and ADMM.

Accelerated Gradient Algorithm

The AG algorithm has attracted extensive attention in the machine learning community due to its optimal convergence rate among all first order techniques and its ability of dealing with large scale data. The general scheme in AG for solving Eq. (6.6) can be described as below: at the k-th iteration, the intermediate (feasible) solution Θ_k can be obtained via

$$\Theta_k = \arg\min_{\Theta} \left(\frac{\gamma_k}{2} \left\| \Theta - \left(\Phi_k - \frac{1}{\gamma_k} \nabla \widehat{S}(\Phi_k) \right) \right\|_F^2 + \alpha \|\Theta\|_* + \beta \|\Theta\|_1 \right), \tag{6.7}$$

where Φ_k denotes a searching point constructed on the intermediate solutions from previous iterations, $\nabla \widehat{S}(\Phi_k)$ denotes the derivative of the loss function in Eq. (6.5) at Φ_k, and γ_k specifies the step size which can be determined by iterative increment until the condition

$$\widehat{S}(\Theta_k) \leq \widehat{S}(\Phi_k) + \langle \nabla f(\Phi_k), \Theta_k - \Phi_k \rangle + \frac{\gamma_k}{2} \|\Theta_k - \Phi_k\|_F^2$$

is satisfied. The operation in Eq. (6.7) is commonly referred to as proximal operator [91], and its efficient computation is critical for the practical convergence of the AG-type algorithm. Next we present an efficient alternating optimization procedure to solve Eq. (6.7) with a given γ_k.

Dual Formulation

The problem in Eq. (6.7) is not easy to solve directly; next we show that this problem can be efficiently solved in its dual form. By reformulating $\|\Theta\|_*$ and $\|\Theta|_1$ into the equivalent dual forms, we convert Eq. (6.7) into a max-min formulation as

$$\begin{aligned} \max_{L,S} \min_{\Theta} \quad & \|\Theta - \widehat{\Phi}\|_F^2 + \widehat{\alpha}\langle L, \Theta\rangle + \widehat{\beta}\langle S, \Theta\rangle \\ \text{subject to} \quad & \|L\|_2 \leq 1, \ \|S\|_\infty \leq 1, \end{aligned} \tag{6.8}$$

where $\widehat{\Phi} = \Phi_k - \nabla \widehat{S}(\Phi_k)/\gamma_k$, $\widehat{\alpha} = 2\alpha/\gamma_k$, and $\widehat{\beta} = 2\beta/\gamma_k$. It can be verified that in Eq. (6.8) the Slater condition is satisfied and strong duality holds [60]. Also the optimal Θ can be expressed as a function of L and S given by

$$\Theta = \widehat{\Phi} - \frac{1}{2}(\widehat{\alpha} L + \widehat{\beta} S). \tag{6.9}$$

By substituting Eq. (6.9) into Eq. (6.8), we obtain the dual form of Eq. (6.7) as

$$\begin{aligned} \min_{L,S} \quad & \|\widehat{\alpha} L + \widehat{\beta} S - 2\widehat{\Phi}\|_F^2 \\ \text{subject to} \quad & \|L\|_2 \leq 1, \ \|S\|_\infty \leq 1. \end{aligned} \tag{6.10}$$

Alternating Optimization

The optimization problem in Eq. (6.10) is smooth convex and it has two optimization variables. For such type of problems, coordinate descent (CD) method is routinely used to compute its globally optimal solution [117]. To solve Eq. (6.10), the CD method alternatively optimizes one of the two variables with the other variable fixed. Our analysis below shows that the variables L and S in Eq. (6.10) can be optimized efficiently. Note that the convergence rate of the CD method is not known, however, it converges very fast in practice (less than 10 iterations in our experiments).

Optimization of L For a given S, the variable L can be optimized via solving the following problem:

$$\begin{aligned} \min_{L} \quad & \|L - \widehat{L}\|_F^2 \\ \text{subject to} \quad & \|L\|_2 \leq 1, \end{aligned} \tag{6.11}$$

where $\widehat{L} = (2\widehat{\Phi} - \widehat{\beta}S)/\widehat{\alpha}$. The optimization on L above can be interpreted as computing an optimal projection of a given matrix over a unit spectral norm ball. Our analysis shows that the optimal solution to Eq. (6.11) can be expressed in an analytic form as summarized in the following theorem.

Theorem 6.3.1. *For arbitrary $\widehat{L} \in \mathbb{R}^{h \times k}$ in Eq. (6.11), denote its SVD by $\widehat{L} = U\Sigma V^T$, where $r =$ rank$(\widehat{L})$, $U \in \mathbb{R}^{h \times r}$, $V \in \mathbb{R}^{k \times r}$, and $\Sigma =$ diag$\left(\sigma_1, \cdots, \sigma_r\right) \in \mathbb{R}^{r \times r}$. Let $\hat{\sigma}_i^* = \min\left(\sigma_i, 1\right), \;\; i = 1, \cdots, r$. Then the optimal solution to Eq. (6.11) is given by*

$$L^* = U\hat{\Sigma}V^T, \;\; \hat{\Sigma} = \textit{diag}\left(\hat{\sigma}_1^*, \cdots, \hat{\sigma}_r^*\right). \tag{6.12}$$

Proof. Assume the existence of a set of left and right singular vector pairs shared by the optimal L^* to Eq. (6.11) and the given $\widehat{L}$ for their non-zero singular values. Under such an assumption, it can be verified that the singular values of L^* can be obtained via

$$\begin{aligned} \min_{\{\hat{\sigma}_i\}} \quad & \left(\hat{\sigma}_i - \sigma_i\right)^2 \\ \text{subject to} \quad & 0 \leq \hat{\sigma}_i \leq 1, \;\; i = 1, \cdots, r, \end{aligned}$$

to which the optimal solution is given by $\hat{\sigma}_i^* = \min(\sigma_i, 1)$ $(\forall i)$; hence the expression of L^* coincides with Eq. (6.12). Therefore, all that remains is to show that our assumption (on the left and right singular vector pairs of L^* and $\widehat{L}$) holds.

Denote the Lagrangian associated with the problem in Eq. (6.11) as $h(L, \lambda) = \|L - \widehat{L}\|_F^2 + \lambda\left(\|L\|_2 - 1\right)$, where λ denotes the dual variable. Since $\mathbf{0}$ is strictly feasible in Eq. (6.11), namely, $\|\mathbf{0}\|_2 < 1$, strong duality holds for Eq. (6.11). Let λ^* be the optimal dual variable to Eq. (6.11). Therefore we have $L^* = \arg\min_L h(L, \lambda^*)$. It is well known that L^* minimizes $h(L, \lambda^*)$ if and only if $\mathbf{0}$ is a subgradient of $h(L, \lambda^*)$ at L^*, i.e.,

$$\mathbf{0} \in 2(L^* - \widehat{L}) + \lambda^* \partial \|L^*\|_2. \tag{6.13}$$

For any matrix Z, the subdifferential of $\|Z\|_2$ is given by [73]

$$\partial\|Z\|_2 = \text{conv}\left\{u_z v_z^T : \|u_z\| = \|v_z\| = 1, Z v_z = \|Z\|_2 u_z\right\},$$

where conv$\{c\}$ denotes the convex hull of the set c. Specifically, any element of $\partial\|Z\|_2$ has the form

$$\sum_i \alpha_i u_{zi} v_{zi}^T, \;\; \alpha_i \geq 0, \;\; \sum_i \alpha_i = 1,$$

where u_{zi} and v_{zi} are any left and right singular vectors of Z corresponding to its largest singular value (the top singular values may share a common value). From Eq. (6.13) and the definition of $\partial\|Z\|_2$, there exist $\{\hat{\alpha}_i\}$ such that $\hat{\alpha}_i > 0,\ \sum_i \hat{\alpha}_i = 1,\ \sum_i \hat{\alpha}_i u_{li} v_{li}^T \in \partial\|L^*\|_2$, and

$$\widehat{L} = L^* + \frac{\lambda^*}{2} \sum_i \hat{\alpha}_i u_{li} v_{li}^T, \tag{6.14}$$

where u_{li} and v_{li}^T correspond to any left and right singular vectors of L^* corresponding to its largest singular value. Since $\lambda^*, \hat{\alpha}_i > 0$, Eq. (6.14) verifies the existence of a set of left and right singular vector pairs shared by L^* and $\widehat{L}$. This completes the proof. □

Optimization of S For a given L, the variable S can be optimized via solving the following problem:

$$\begin{aligned} \min_{S} \quad & \|S - \widehat{S}\|_F^2 \\ \text{subject to} \quad & \|S\|_\infty \leq 1, \end{aligned} \tag{6.15}$$

where $\widehat{S} = (2\widehat{\Phi} - \widehat{\alpha}L)/\widehat{\beta}$. Similarly, the optimization on S can be interpreted as computing a projection of a given matrix over an infinity norm ball. It also admits an analytic solution as summarized in the following theorem.

Lemma 6.3.1. *For any matrix $\widehat{S}$, the optimal solution to Eq. (6.15) is given by*

$$S^* = \textit{sgn}(\widehat{S}) \circ \min(|\widehat{S}|, \boldsymbol{1}), \tag{6.16}$$

*where $\circ$ denotes the component-wise multiplication operator, and **1** denotes the matrix with entries 1 of appropriate size.*

Alternating Direction Method of Multipliers

The ADMM algorithm [115] is suitable for dealing with non-smooth (convex) optimizations problems, as it blends the decomposability of dual ascent with the superior convergence of the method of multipliers. We present two implementations of the ADMM algorithm for solving Eq. (6.6). The key difference lies in the use of different numbers of auxiliary variables to separate the smooth components from the non-smooth components of the objective function in Eq. (6.6).

The First Implementation: ADMM1

By adding an auxiliary variable Ψ, we reformulate Eq. (6.6) as

$$\begin{aligned} \min_{\Theta,\Psi} \quad & \widehat{S}(\Theta) + \alpha\|\Psi\|_* + \beta\|\Theta\|_1 \\ \text{subject to} \quad & \Theta = \Psi. \end{aligned} \tag{6.17}$$

The augmented Lagrangian of Eq. (6.17) can be expressed as

$$\mathcal{L}_\rho^1(\Theta, \Psi, \Gamma) = \widehat{S}(\Theta) + \alpha\|\Psi\|_* + \beta\|\Theta\|_1 + \langle \Theta - \Psi, \Gamma \rangle + \frac{\rho}{2}\|\Theta - \Psi\|_F^2. \tag{6.18}$$

To solve Eq. (6.17), ADMM1 consists of the following iterations:

$$\begin{aligned} \Theta_{k+1} &= \arg\min_{\Theta} \mathcal{L}_\rho^1(\Theta, \Psi_k, \Gamma_k), & (6.19)\\ \Psi_{k+1} &= \arg\min_{\Psi} \mathcal{L}_\rho^1(\Theta_{k+1}, \Psi, \Gamma_k), & (6.20)\\ \Gamma_{k+1} &= \Gamma_k + \rho\left(\Theta_{k+1} - \Psi_{k+1}\right), & (6.21) \end{aligned}$$

where Θ_k, Ψ_k, and Γ_k denote the intermediate solutions of ADMM1 at the k-th iteration, and ρ is a pre-specified constant.

Specifically, if we employ the least squares loss, i.e., $\widehat{S}(\Theta) = \|\mathcal{G}_X\Theta - Y\|_F^2/N$, the optimization problems in Eqs. (6.19) and (6.21) can be efficiently solved as below.

Update on Θ The optimal Θ_{k+1} to Eq. (6.19) can be obtained via

$$\Theta_{k+1} = \arg\min_{\Theta} \left(\frac{1}{N}\|\mathcal{G}_X\Theta - Y\|_F^2 + \beta\|\Theta\|_1 + \langle \Theta, \Gamma_k \rangle + \frac{\rho}{2}\|\Theta - \Psi_k\|_F^2 \right), \tag{6.22}$$

which can be efficiently solved via the gradient-type methods [71, 116].

Update on Ψ The optimal Ψ_{k+1} to Eq. (6.20) can be obtained via

$$\Psi_{k+1} = \arg\min_{\Psi} \left(\alpha\|\Psi\|_* - \langle \Psi, \Gamma_k \rangle + \frac{\rho}{2}\|\Theta_{k+1} - \Psi\|_F^2 \right).$$

The optimization problem above admits an analytical solution [94]. Assume rank $(\Theta_{k+1} + \Gamma_k/\rho) = r$. Let $\Theta_{k+1} + \Gamma_k/\rho = U_r\Sigma_r V_r^T$ be the singular value decomposition of $\Theta_{k+1} + \Gamma_k/\rho$, where U_r and V_r consist of respectively r orthonormal columns, and $\Sigma_r =$ diag $\{(\sigma_1, \sigma_2, \cdots, \sigma_r)\}$. Then the optimal Ψ_{k+1} is given by

$$\Psi_{k+1} = U_r\hat{\Sigma}V_r^T, \ \hat{\Sigma} = \text{diag}\left\{ \left(\sigma_i - \frac{\alpha}{\rho} \right)_+ \right\}, \tag{6.23}$$

where $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise.

The Second Implementation: ADMM2

By adding two auxiliary variables Ψ^1 and Ψ^2, we reformulate Eq. (6.6) as

$$\begin{aligned} \min_{\Theta, \Psi^1, \Psi^2} \quad & \widehat{S}(\Theta) + \alpha\|\Psi^1\|_* + \beta\|\Psi^2\|_1 \\ \text{subject to} \quad & \Theta = \Psi^1, \ \Theta = \Psi^2. \end{aligned} \tag{6.24}$$

Similarly, the augmented Lagrangian of Eq. (6.24) can be expressed as

$$\begin{aligned}&\mathcal{L}^2_{\rho_1,\rho_2}(\Theta, \Psi^1, \Psi^2, \Gamma^1, \Gamma^2)\\ =\ & \widehat{S}(\Theta) + \alpha\|\Psi^1\|_* + \beta\|\Psi^2\|_1 + \langle \Theta - \Psi^1, \Gamma^1\rangle + \langle \Theta - \Psi^2, \Gamma_2\rangle + \frac{\rho_1}{2}\|\Theta - \Psi^1\|_F^2 + \frac{\rho_2}{2}\|\Theta - \Psi^2\|_F^2.\end{aligned}$$

To solve Eq. (6.24), ADMM2 consists of the following iterations:

$$\begin{aligned}\Theta_{k+1} &= \arg\min_{\Theta} \mathcal{L}^2_{\rho_1,\rho_2}(\Theta, \Psi^1_k, \Psi^2_k, \Gamma^1_k, \Gamma^2_k), && (6.25)\\ \left(\Psi^1_{k+1}, \Psi^2_{k+1}\right) &= \arg\min_{\Psi^1,\Psi^2} \mathcal{L}^2_{\rho_1,\rho_2}(\Theta_{k+1}, \Psi^1, \Psi^2, \Gamma^1_k, \Gamma^2_k), && (6.26)\\ \Gamma^1_{k+1} &= \Gamma^1_k + \rho_1\left(\Theta_{k+1} - \Psi^1_{k+1}\right), && (6.27)\\ \Gamma^2_{k+1} &= \Gamma^2_k + \rho_2\left(\Theta_{k+1} - \Psi^2_{k+1}\right), && (6.28)\end{aligned}$$

where Θ_k, Ψ^1_k, Ψ^2_k, Γ^1_k, and Γ^2_k denote the intermediate solutions at the k-th iteration of the ADMM2 method.

Specifically, if we employ $\widehat{S}(\Theta) = \|\mathcal{G}_X\Theta - Y\|_F^2/N$ as the loss function in Eq. (6.24), the optimization problems in Eqs. (6.25), (6.26), (6.27), and (6.28) can be efficiently solved as below.

Update on Θ The optimal Θ_{k+1} to Eq. (6.25) can be obtained via

$$\Theta_{k+1} = \arg\min_{\Theta}\left(\frac{1}{N}\|\mathcal{G}_X\Theta - Y\|_F^2 + \langle\Theta, \Gamma^1_k + \Gamma^2_k\rangle + \frac{\rho_1}{2}\|\Theta - \Psi^1_k\|_F^2 + \frac{\rho_2}{2}\|\Theta - \Psi^2_k\|_F^2\right).$$

Note that the optimal Θ_{k+1} can be obtained via solving a systems of linear equations.

Update on Ψ^1 and Ψ^2 The optimal Ψ^1_{k+1} and Ψ^1_{k+1} to Eq. (6.26) can be obtained via

$$\begin{aligned}\Psi^1_{k+1} &= \arg\min_{\Psi^1}\left(\alpha\|\Psi^1\|_* - \langle\Psi^1, \Gamma^1_k\rangle + \frac{\rho_1}{2}\|\Theta_{k+1} - \Psi^1\|_F^2\right), && (6.29)\\ \Psi^2_{k+1} &= \arg\min_{\Psi^2}\left(\beta\|\Psi^2\|_1 - \langle\Psi^2, \Gamma^2_k\rangle + \frac{\rho_2}{2}\|\Theta_{k+1} - \Psi^2\|_F^2\right). && (6.30)\end{aligned}$$

It can be verified that Eq. (6.29) admits an analytical solution. Assume rank $\left(\Theta_{k+1} + \Gamma^1_k/\rho_1\right) = r$. Let $\Theta_{k+1} + \Gamma^1_k/\rho_1 = U_r\Sigma_r V_r^T$ be the singular value decomposition of $\Theta_{k+1} + \Gamma^1_k/\rho_1$, where U_r and V_r consist of respectively r orthonormal columns, and $\Sigma_r = \text{diag}\left\{(\sigma_1, \sigma_2, \cdots, \sigma_r)\right\}$. Then the optimal Ψ^1_{k+1} is given by

$$\Psi^1_{k+1} = U_r\hat{\Sigma}V_r^T,\ \hat{\Sigma} = \text{diag}\left\{\left(\sigma_i - \frac{\alpha}{\rho_1}\right)_+\right\}, \qquad (6.31)$$

where $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise. Moreover, it can also be verified that Eq. (6.30) admits an analytical solution. Let ψ, θ, and γ be the entries of Ψ^2_{k+1}, Θ_{k+1}, and Γ^2_k at the same coordinates. The optimal ψ is given by

$$\psi = \begin{cases} \theta + \frac{1}{\rho_2}(\gamma - \beta) & \theta + \frac{1}{\rho_2}\gamma > \frac{1}{\rho_2}\beta \\ 0 & -\frac{1}{\rho_2}\beta \leq \theta + \frac{1}{\rho_2}\gamma \leq \frac{1}{\rho_2}\beta \\ \theta + \frac{1}{\rho_2}(\gamma + \beta) & \theta + \frac{1}{\rho_2}\gamma < -\frac{1}{\rho_2}\beta \end{cases} . \tag{6.32}$$

6.4 Theoretical Analysis

In this section, we present a performance bound for the function estimation scheme in Eq. (6.3). Such a performance bound measures how well the estimation scheme can approximate the regression functions $\{f_j\}$ in Eq. (6.2) via the sparse low-rank coefficient Θ.

Basic Properties of the Optimal Solution

We first define two operators, namely $\mathcal{S}_0$ and $\mathcal{S}_1$, on an arbitrary matrix pair (of the same size) based on Lemma 3.4 in [94], as summarized in the following lemma.

Lemma 6.4.1. *Given any Θ and Δ of size $h \times k$, let rank$(\Theta) = r$ and denote the SVD of Θ as*

$$\Theta = U \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V^T,$$

where $U \in \mathbb{R}^{h\times h}$ and $V \in \mathbb{R}^{k\times k}$ are orthogonal, and $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal consisting of the non-zero singular values on its main diagonal. Let

$$\widehat{\Delta} = U^T \Delta V = \begin{bmatrix} \widehat{\Delta}_{11} & \widehat{\Delta}_{12} \\ \widehat{\Delta}_{21} & \widehat{\Delta}_{22} \end{bmatrix},$$

where $\widehat{\Delta}_{11} \in \mathbb{R}^{r\times r}$, $\widehat{\Delta}_{12} \in \mathbb{R}^{r\times(k-r)}$, $\widehat{\Delta}_{21} \in \mathbb{R}^{(h-r)\times r}$, and $\widehat{\Delta}_{22} \in \mathbb{R}^{(h-r)\times(k-r)}$. Define $\mathcal{S}_0$ and $\mathcal{S}_1$ as

$$\mathcal{S}_0(\Theta, \Delta) = U \begin{bmatrix} \widehat{\Delta}_{11} & \widehat{\Delta}_{12} \\ \widehat{\Delta}_{21} & \mathbf{0} \end{bmatrix} V^T, \ \mathcal{S}_1(\Theta, \Delta) = U \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widehat{\Delta}_{22} \end{bmatrix} V^T.$$

Then the following conditions hold: rank$(\mathcal{S}_0(\Theta, \Delta)) \leq 2r$, $\Theta\mathcal{S}_1(\Theta, \Delta)^T = 0$, $\Theta^T\mathcal{S}_1(\Theta, \Delta) = 0$.

The result presented in Lemma 6.4.1 implies a condition under which the trace norm on a matrix pair is additive. From Lemma 6.4.1 we can easily verify that

$$\|\Theta + \mathcal{S}_1(\Theta, \Delta)\|_* = \|\Theta\|_* + \|\mathcal{S}_1(\Theta, \Delta)\|_*, \tag{6.33}$$

for arbitrary Θ and Δ of the same size. To avoid clutter notation, we denote $\mathcal{S}_0(\Theta, \Delta)$ by $\mathcal{S}_0(\Delta)$, and $\mathcal{S}_1(\Theta, \Delta)$ by $\mathcal{S}_1(\Delta)$ throughout this chapter, as the appropriate Θ can be easily determined from the context.

As a consequence of Lemma 6.4.1, we derive a bound on the trace norm of the matrices of interest as summarized below.

Corollary 6.4.1. *Given an arbitrary matrix pair $\widehat{\Theta}$ and Θ, let $\Delta = \widehat{\Theta} - \Theta$. Then*

$$\|\widehat{\Theta} - \Theta\|_* + \|\Theta\|_* - \|\widehat{\Theta}\|_* \leq 2\|\mathcal{S}_0(\Delta)\|_*.$$

Proof. From Lemma 6.4.1 we have $\Delta = \mathcal{S}_0(\Delta) + \mathcal{S}_1(\Delta)$ for the matrix pair Θ and Δ. Moreover,

$$\begin{aligned} \|\widehat{\Theta}\|_* &= \|\Theta + \mathcal{S}_0(\Delta) + \mathcal{S}_1(\Delta)\|_* \geq \|\Theta + \mathcal{S}_1(\Delta)\|_* - \|\mathcal{S}_0(\Delta)\|_* \\ &= \|\Theta\|_* + \|\mathcal{S}_1(\Delta)\|_* - \|\mathcal{S}_0(\Delta)\|_*, \end{aligned} \tag{6.34}$$

where the inequality above follows from the triangle inequality and the last equality above follows from Eq. (6.33). Using the result in Eq. (6.34), we have

$$\begin{aligned} \|\widehat{\Theta} - \Theta\|_* + \|\Theta\|_* - \|\widehat{\Theta}\|_* &\leq \|\Delta\|_* + \|\Theta\|_* - \|\Theta\|_* - \|\mathcal{S}_1(\Delta)\|_* + \|\mathcal{S}_0(\Delta)\|_* \\ &\leq 2\|\mathcal{S}_0(\Delta)\|_*. \end{aligned}$$

We complete the proof of this corollary. □

Analogous to the bound on the trace norm in Corollary 6.4.1, we also derive a bound on the ℓ_1-norm of the matrices of interest in the following lemma. For arbitrary matrices Θ and Δ, we denote by $J(\Theta) = \{(i,j)\}$ the coordinate set (the location set of nonzero entries) of Θ, and by $J(\Theta)_\perp$ the associated complement (the location set of zero entries); we denote by $\Delta_{J(\Theta)}$ the matrix of the same entries as Δ on the set $J(\Theta)$ and of zero entries on the set $J(\Theta)_\perp$. We now present a result associated with $J(\Theta)$ and $J(\Theta)_\perp$ in the following lemma. Note that a similar result for the vector case is presented in [95].

Lemma 6.4.2. *Given a matrix pair $\widehat{\Theta}$ and Θ of the same size, the inequality below always holds*

$$\|\widehat{\Theta} - \Theta\|_1 + \|\Theta\|_1 - \|\widehat{\Theta}\|_1 \leq 2\|\widehat{\Theta}_{J(\Theta)} - \Theta_{J(\Theta)}\|_1. \tag{6.35}$$

Proof. It can be verified that the inequality

$$\|\Theta_{J(\Theta)}\|_1 - \|\widehat{\Theta}_{J(\Theta)}\|_1 \leq \|(\widehat{\Theta} - \Theta)_{J(\Theta)}\|_1$$

and the equalities

$$\Theta_{J(\Theta)_\perp} = \mathbf{0},\ \ \|(\widehat{\Theta}-\Theta)_{J(\Theta)_\perp}\|_1 - \|\widehat{\Theta}_{J(\Theta)}\|_1 = \mathbf{0}$$

hold. Therefore we can derive

$$\begin{aligned}
& \|\widehat{\Theta}-\Theta\|_1 + \|\Theta\|_1 - \|\widehat{\Theta}\|_1 \\
= \ & \|(\widehat{\Theta}-\Theta)_{J(\Theta)}\|_1 + \|(\widehat{\Theta}-\Theta)_{J(\Theta)_\perp}\|_1 + \|\Theta_{J(\Theta)}\|_1 + \|\Theta_{J(\Theta)_\perp}\|_1 - \|\widehat{\Theta}_{J(\Theta)}\|_1 - \|\widehat{\Theta}_{J(\Theta)_\perp}\|_1 \\
\leq \ & 2\|(\widehat{\Theta}-\Theta)_{J(\Theta)}\|_1.
\end{aligned}$$

This completes the proof of this lemma. □

We present a concentration inequality, which is important for our following analysis.

Lemma 6.4.3. *Let $\sigma_{X(l)}$ be the maximum singular value of the matrix $\mathcal{G}_X \in \mathbb{R}^{n\times h}$; let $W \in \mathbb{R}^{n\times k}$ be the matrix of i.i.d entries as $w_{ij} \sim \mathcal{N}(0, \sigma_w^2)$. Let $\lambda = 2\sigma_{X(l)}\sigma_w\sqrt{n}\left(1+\sqrt{k/n}+t\right)/N$. Then*

$$\Pr\left(\|W^T\mathcal{G}_X\|_2/N \leq \lambda/2\right) \geq 1 - \exp\left(-nt^2/2\right).$$

Proof. It is known [118] that a Gaussian matrix $\widehat{W} \in \mathbb{R}^{n\times k}$ with $n \geq k$ and $\hat{w}_{ij} \sim \mathcal{N}(0, 1/n)$ satisfies

$$\text{Pr}\left(\|\widehat{W}\|_2 > 1+\sqrt{k/n}+t\right) \leq \exp\left(-nt^2/2\right), \tag{6.36}$$

where t is a universal constant. From the definition of the largest singular value, there exist a vector $b \in \mathbb{R}^h$ of length 1, i.e., $\|b\|_2 = 1$, such that $\|W^T\mathcal{G}_X\|_2 = \|W^T\mathcal{G}_X b\|_2 \leq \|W\|_2\|\mathcal{G}_X b\|_2 \leq \sigma_{X(l)}\|W\|_2$. Since $w_{ij}/\left(\sigma_w\sqrt{n}\right) \sim \mathcal{N}(0, 1/n)$, we have

$$\text{Pr}\left(\left\|W^T\mathcal{G}_X\right\|_2/N > \lambda/2\right) \leq \text{Pr}\left(\sigma_{X(l)}\left\|W\right\|_2/N > \lambda/2\right).$$

Applying the result in Eq. (6.36) into the inequality above, we complete the proof of this lemma. □

We present some basic properties of the optimal solution defined in Eq. (6.6); these properties are important building blocks of our following theoretical analysis.

Lemma 6.4.4. *Consider the optimization problem in Eq. (6.6) for $h, k \geq 2$ and $n \geq 1$. Given n sample pairs as $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times d}$ and $Y = [y_1, \cdots, y_n]^T \in \mathbb{R}^{n\times k}$. Let $\mathcal{F}$ and $\mathcal{G}_X$ be defined in Eq. (6.2) and Eq. (6.4), respectively; let $\sigma_{X(l)}$ be the largest singular values of $\mathcal{G}_X$. Assume that $W \in \mathbb{R}^{n\times k}$ has independent and identically distributed (i.i.d.) entries as $w_{ij} \sim \mathcal{N}(0, \sigma_w^2)$. Take*

$$\alpha + \beta = \frac{2\sigma_{X(l)}\sigma_w\sqrt{n}}{N}\left(1+\sqrt{\frac{k}{n}}+t\right), \tag{6.37}$$

where $N = n \times k$ and t is a universal constant. Then with probability of at least $1 - \exp\left(-nt^2/2\right)$, for the minimizer $\widehat{\Theta}$ in Eq. (6.6) and any $\Theta \in \mathbb{R}^{h\times k}$, we have

$$\frac{1}{N}\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2 \le \frac{1}{N}\|\mathcal{G}_X\Theta - \mathcal{F}\|_F^2 + 2\alpha\|\mathcal{S}_0(\widehat{\Theta} - \Theta)\|_* + 2\beta\|(\widehat{\Theta} - \Theta)_{J(\Theta)}\|_1, \tag{6.38}$$

where $\mathcal{S}_0$ is an operator defined in Lemma 6.4.1 of the supplemental material.

Proof. From the definition of $\widehat{\Theta}$ in Eq. (6.6), we have

$$\widehat{S}(\widehat{\Theta}) + \alpha\|\widehat{\Theta}\|_* + \beta\|\widehat{\Theta}\|_1 \le \widehat{S}(\Theta) + \alpha\|\Theta\|_* + \beta\|\Theta\|_1.$$

By substituting $Y = \mathcal{F} + W$ and Eq. (6.5) into the previous inequality, we have

$$\begin{aligned} & \frac{1}{N}\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2 \\ \le \; & \frac{1}{N}\|\mathcal{G}_X\Theta - \mathcal{F}\|_F^2 + \frac{2}{N}\langle W, \mathcal{G}_X(\widehat{\Theta} - \Theta)\rangle + \alpha\left(\|\Theta\|_* - \|\widehat{\Theta}\|_*\right) + \beta\left(\|\Theta\|_1 - \|\widehat{\Theta}\|_1\right). \end{aligned}$$

Define the random event

$$\mathcal{A} = \left\{\frac{1}{N}\|\mathcal{G}_X^T W\|_2 \le \frac{\alpha + \beta}{2}\right\}. \tag{6.39}$$

Taking $\alpha + \beta$ as the value in Eq. (6.37), it follows from Lemma 6.4.3 of the supplemental materia that $\mathcal{A}$ holds with probability of at least $1 - \exp\left(-\frac{nt^2}{2}\right)$. Therefore, we have

$$\begin{aligned} \langle W, \mathcal{G}_X(\widehat{\Theta} - \Theta)\rangle &= \frac{\alpha+\beta}{\alpha+\beta}\langle W, \mathcal{G}_X(\widehat{\Theta} - \Theta)\rangle \\ \le \; & \frac{\alpha}{\alpha+\beta}\|\mathcal{G}_X^T W\|_2\|\widehat{\Theta} - \Theta\|_* + \frac{\beta}{\alpha+\beta}\|\mathcal{G}_X^T W\|_\infty\|\widehat{\Theta} - \Theta\|_1 \\ \le \; & \frac{N}{2}\left(\alpha\|\widehat{\Theta} - \Theta\|_* + \beta\|\widehat{\Theta} - \Theta\|_1\right), \end{aligned}$$

where the second inequality follows from $\|\mathcal{G}_X^T W\|_2 \ge \|\mathcal{G}_X^T W\|_\infty$. Therefore, under $\mathcal{A}$, we have

$$\begin{aligned} & \frac{1}{N}\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2 \\ \le & \frac{1}{N}\|\mathcal{G}_X\Theta - \mathcal{F}\|_F^2 + \alpha\|\widehat{\Theta} - \Theta\|_* + \beta\|\widehat{\Theta} - \Theta\|_1 + \alpha\left(\|\Theta\|_* - \|\widehat{\Theta}\|_*\right) + \beta\left(\|\Theta\|_1 - \|\widehat{\Theta}\|_1\right). \end{aligned}$$

From Corollary 6.4.1 and Lemma 6.4.2 of the supplemental material, we complete the proof. □

Main Assumption

We introduce a key assumption on the dictionary of basis functions $\mathcal{G}_X$. Based on such an assumption, we derive a performance bound for the sparse trace norm regularization formulation in Eq. (6.6).

Assumption 6.4.1. *For a matrix pair Θ and Δ of size $h \times k$, let $s \leq \min(h,k)$ and $q \leq h \times k$. We assume that there exist constants $\kappa_1(s)$ and $\kappa_2(q)$ such that*

$$\kappa_1(s) \triangleq \min_{\Delta \in \mathcal{R}(s,q)} \frac{\|\mathcal{G}_X \Delta\|_F}{\sqrt{N}\|\mathcal{S}_0(\Delta)\|_*} > 0, \ \kappa_2(q) \triangleq \min_{\Delta \in \mathcal{R}(s,q)} \frac{\|\mathcal{G}_X \Delta\|_F}{\sqrt{N}\|\Delta_{J(\Theta)}\|_1} > 0, \tag{6.40}$$

where the restricted set $\mathcal{R}(s,q)$ is defined as

$$\mathcal{R}(s,q) = \left\{\Delta \in \mathbb{R}^{h \times k}, \Theta \in \mathbb{R}^{h \times k} \,|\, \Delta \neq 0, \ \mathit{rank}(\mathcal{S}_0(\Delta)) \leq s, \ |J(\Theta)| \leq q\right\},$$

and $|J(\Theta)|$ denotes the number of nonzero entries in the matrix Θ.

Our assumption on $\kappa_1(s)$ in Eq. (6.40) is closely related to but less restrictive than the RSC condition used in [100]; its denominator is only a part of the one in RSC and in a different matrix norm as well. Our assumption on $\kappa_2(q)$ is similar to the RE condition used in [95] except that its denominator is in a different matrix norm; our assumption can also be implied by sufficient conditions similar to the ones in [95].

Performance Bound

We derive a performance bound for the sparse trace norm structure obtained by solving Eq. (6.6). This bound measures how well the optimal $\widehat{\Theta}$ can be used to approximate $\mathcal{F}$ by evaluating the averaged estimation error, i.e., $\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2/N$.

Theorem 6.4.1. *Consider the optimization problem in Eq. (6.6) for $h,k \geq 2$ and $n \geq 1$. Given n sample pairs as $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n \times d}$ and $Y = [y_1, \cdots, y_n]^T \in \mathbb{R}^{n \times k}$, let $\mathcal{F}$ and $\mathcal{G}_X$ be defined in Eqs. (6.2) and (6.4), respectively; let $\sigma_{X(l)}$ be the largest singular value of $\mathcal{G}_X$. Assume that $W \in \mathbb{R}^{n \times k}$ has i.i.d. entries as $w_{ij} \sim \mathcal{N}(0, \sigma_w^2)$. Take $\alpha + \beta$ as the value in Eq. (6.37). Then with probability of at least $1 - \exp\left(-nt^2/2\right)$, for the minimizer $\widehat{\Theta}$ in Eq. (6.6), we have*

$$\frac{1}{N}\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2 \leq (1+\epsilon)\inf_{\Theta}\left\{\frac{1}{N}\|\mathcal{G}_X\Theta - \mathcal{F}\|_F^2\right\} + \mathcal{E}(\epsilon)\left(\frac{\alpha^2}{\kappa_1^2(2r)} + \frac{\beta^2}{\kappa_2^2(c)}\right), \tag{6.41}$$

where $\inf$ is taken over all $\Theta \in \mathbb{R}^{h \times k}$ with rank(Θ) $\leq r$ and $|J(\Theta)| \leq c$, and $\mathcal{E}(\epsilon) > 0$ is a constant depending only on ϵ.

Proof. Denote $\Delta = \widehat{\Theta} - \Theta$ in Eq. (6.38). We have

$$\frac{1}{N}\|\mathcal{G}_X\widehat{\Theta} - \mathcal{F}\|_F^2 \leq \frac{1}{N}\|\mathcal{G}_X\Theta - \mathcal{F}\|_F^2 + 2\alpha\|\mathcal{S}_0(\Delta)\|_* + 2\beta\|\Delta_{J(\Theta)}\|_1. \tag{6.42}$$

Given $\mathcal{S}_0(\Delta) \leq 2r$ (from Lemma 6.4.1 of the supplemental material) and $|J(\Theta)| \leq c$, we derive upper bounds on the components $2\alpha\|\mathcal{S}_0(\Delta)\|_*$ and $2\beta\|\Delta_{J(\Theta)}\|_1$ over the restrict set $\mathcal{R}(2r, c)$ based on Assumptions 6.4.1, respectively. It follows that

$$\begin{aligned}2\alpha\|\mathcal{S}_0(\Delta)\|_* &\le \frac{2\alpha}{\kappa_1(2r)\sqrt{N}}\|\mathcal{G}_X(\widehat{\Theta}-\Theta)\|_F \le \frac{2\alpha}{\kappa_1(2r)\sqrt{N}}\left(\|\mathcal{G}_X\widehat{\Theta}-\mathcal{F}\|_F+\|\mathcal{G}_X\Theta-\mathcal{F}\|_F\right)\\ &\le \frac{\alpha^2\tau}{\kappa_1^2(2r)}+\frac{1}{N\tau}\|\mathcal{G}_X\widehat{\Theta}-\mathcal{F}\|_F^2+\frac{\alpha^2\tau}{\kappa_1^2(2r)}+\frac{1}{N\tau}\|\mathcal{G}_X\Theta-\mathcal{F}\|_F^2, \qquad (6.43)\end{aligned}$$

where the last inequality above follows from $2ab \le a^2\tau + b^2/\tau$ for $\tau > 0$. Similarly, we have

$$2\beta\|\Delta_{J(\Theta)}\|_1 \le \frac{\beta^2\tau}{\kappa_2^2(c)}+\frac{1}{N\tau}\|\mathcal{G}_X\widehat{\Theta}-\mathcal{F}\|_F^2+\frac{\beta^2\tau}{\kappa_2^2(c)}+\frac{1}{N\tau}\|\mathcal{G}_X\Theta-\mathcal{F}\|_F^2. \qquad (6.44)$$

Substituting Eqs. (6.43) and (6.44) into Eq. (6.42), we have

$$\frac{1}{N}\|\mathcal{G}_X\widehat{\Theta}-\mathcal{F}\|_F^2 \le \frac{\tau+2}{(\tau-2)N}\|\mathcal{G}_X\Theta-\mathcal{F}\|_F^2+\frac{2\tau^2}{\tau-2}\left(\frac{\alpha^2}{\kappa_1^2(2r)}+\frac{\beta^2}{\kappa_2^2(c)}\right).$$

Setting $\tau = 2 + 4/\epsilon$ and $\mathcal{E}(\epsilon) = 2(\epsilon+2)^2/\epsilon$ in the inequality above, we complete the proof. □

By choosing specific values for α and β, we can refine the performance bound described in Eq. (6.41). It follows from Eq. (6.37) we have

$$\min_{\alpha,\beta,\alpha+\beta=\gamma}\left(\frac{\alpha^2}{\kappa_1^2(2r)}+\frac{\beta^2}{\kappa_2^2(c)}\right)=\frac{\gamma^2}{\kappa_1^2(2r)+\kappa_2^2(c)},\ \gamma=\frac{2\sigma_{X(l)}\sigma_w\sqrt{n}}{N}\left(1+\sqrt{\frac{k}{n}}+t\right), \qquad (6.45)$$

where the equality of the first equation is achieved by setting α and β proportional to $\kappa_1^2(2r)$ and $\kappa_2^2(q)$, i.e., $\alpha = \gamma\kappa_1^2(2r)/\left(\kappa_1^2(2r)+\kappa_2^2(c)\right)$ and $\beta = \gamma\kappa_2^2(c)/\left(\kappa_1^2(2r)+\kappa_2^2(c)\right)$. Thus the performance bound in Eq. (6.41) can be refined as

$$\frac{1}{N}\|\mathcal{G}_X\widehat{\Theta}-\mathcal{F}\|_F^2 \le (1+\epsilon)\inf_{\Theta}\left\{\frac{1}{N}\|\mathcal{G}_X\Theta-\mathcal{F}\|_F^2\right\}+\frac{4\mathcal{E}(\epsilon)\sigma_{X(l)}^2\sigma_w^2 n}{N^2\left(\kappa_1^2(2r)+\kappa_2^2(c)\right)}\left(1+\sqrt{\frac{k}{n}}+t\right)^2.$$

Note that the performance bound above is independent of the value of α and β, and it is tighter than the one described in Eq. (6.41).

6.5 Experiments

In this section, we evaluate the effectiveness of the sparse trace norm regularization formulation in Eq. (6.6) on benchmark data sets; we also conduct numerical studies on the convergence of AG and two ADMM implementations including ADMM1 and ADMM2 for solving Eq. (6.6) and the convergence of the alternating optimization algorithm for solve Eq. (6.10). Note that we use the least square loss for the following experiments.

Performance Evaluation

We apply the sparse trace norm regularization formulation (S.TraceNorm) on multi-label classification problems, in comparison with the trace norm regularization formulation (TraceNorm) and the

Table 6.1: Averaged performance (with standard derivation) comparison in terms of AUC, Macro F1, and Micro F1. Note that n, d, and m denote the sample size, dimensionality, and label number, respectively.

Data Set (n, d, m)		Business $(9968, 16621, 17)$	Arts $(7441, 17973, 19)$	Health $(9109, 18430, 14)$	Scene $(2407, 294, 6)$
AUC	S.TraceNorm	85.42 ± 0.31	76.31 ± 0.15	86.18 ± 0.56	91.54 ± 0.18
	TraceNorm	83.43 ± 0.41	75.90 ± 0.27	85.24 ± 0.42	90.33 ± 0.24
	OneNorm	81.95 ± 0.26	70.47 ± 0.18	83.60 ± 0.32	88.42 ± 0.31
Macro F1	S.TraceNorm	48.83 ± 0.13	32.83 ± 0.25	60.05 ± 0.36	51.65 ± 0.33
	TraceNorm	47.24 ± 0.15	31.90 ± 0.31	58.91 ± 0.24	50.59 ± 0.08
	OneNorm	46.28 ± 0.25	31.03 ± 0.46	58.01 ± 0.18	46.57 ± 1.10
Micro F1	S.TraceNorm	78.26 ± 0.71	42.91 ± 0.27	67.22 ± 0.47	52.83 ± 0.35
	TraceNorm	78.84 ± 0.11	42.08 ± 0.11	66.92 ± 0.42	52.06 ± 0.49
	OneNorm	78.16 ± 0.17	40.64 ± 0.52	66.37 ± 0.19	47.32 ± 0.13

ℓ_1-norm regularization formulation (OneNorm). AUC, Macro F1, and Micro F1 are used as the classification performance measures. Four benchmark data sets, including Business, Arts, and Health from Yahoo webpage data sets [77] and Scene from LIBSVM multi-label data sets[1], are employed in this experiment. The reported experimental results are averaged over 10 random repetitions of the data sets into training and test sets of the ratio $1 : 9$. We use the AG method to solve the S.TraceNorm formulation, and stop the iterative procedure of AG if the change of the objective values in two successive iterations is smaller than 10^{-8} or the iteration numbers larger than 10^5. The regularization parameters α and β are determined via double cross-validation from the set $\{10^{-2} \times i\}_{i=1}^{10} \cup \{10^{-1} \times i\}_{i=2}^{10} \cup \{2 \times i\}_{i=1}^{10}$.

We present the averaged performance of the competing algorithms in Table 6.1. The main observations are summarized as follows: (1) S.TraceNorm achieves the best performance on all benchmark data sets (except on Business data) in this experiment; this result demonstrates the effectiveness of the induced sparse low-rank structure for multi-label classification tasks; (2) TraceNorm outperforms OneNorm on all benchmark data sets; this result demonstrates the effectiveness of modeling a shared low-rank structure for high-dimensional text and image data analysis.

Numerical Study

We study the practical convergence of AG and ADMM2 by solving Eq. (6.6) on Scene data. In our experiments, we observe that ADMM1 is much slower than ADMM2 and we thus only focus on ADMM2. Note that in AG, we set $\alpha = 1, \beta = 1$; in ADMM2, we set $\alpha = 1$, $\beta = 1$, $\rho_1 = \rho_2 = 10$. For other parameter settings, we observe similar trends.

[1] http://www.csie.ntu.edu.tw/~cjlin

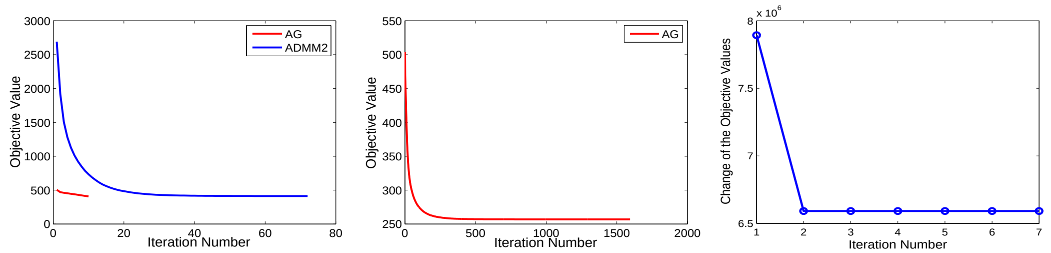


Figure 6.1: Convergence comparison of AG and ADMM2 for solving Eq. (6.6) (left plot); convergence plot of AG for solving Eq. (6.6) (middle plot); and the alternating optimization algorithm for solving the dual formulation of the proximal operator in Eq. (6.10) (right plot).

In the first experiment, we compare AG and ADMM2 in term of the practical convergence. We stop ADMM2 when the change of the objective values in two successive iterations smaller than 10^{-4}; the attained objective value in ADMM2 is used as the stopping criterion for AG, that is, we stop AG if the attained objective value in AG is equal to or smaller than that objective value attained in ADMM2. The convergence curves of ADMM2 and AG are presented in the left plot of Figure 6.1. Clearly, we can observe that AG converges much faster than ADMM2. In the second experiment, we study the convergence of AG. We stop AG when the change of the objective values in two successive iterations smaller than 10^{-8}. The convergence curves is presented in the middle plot of Figure 6.1. We observe that AG converges very fast, and its convergence speed is consistent with the theoretical convergence analysis in [45].

We also conduct numerical study on the alternating optimization algorithm (in Section 6.3) for solving the dual formulation of the proximal operator in Eq. (6.10). Similarly, the alternating optimization algorithm is stopped when the change of the objective values in two successive iterations smaller than 10^{-8}. For illustration, in Eq. (6.10) we randomly generate the matrix $\widehat{\Phi}$ of size 10000 by 5000 from $\mathcal{N}(0,1)$; we then apply the alternating optimization algorithm to solve Eq. (6.10) and plot its convergence curve in the right plot of Figure 6.1. Our experimental results show that the alternating optimization algorithm generally converges within 10 iterations and our results demonstrate the practical efficiency of this algorithm.

6.6 Summary

We study the problem of estimating multiple predictive functions simultaneously in the nonparametric regression setting. In our estimation scheme, each predictive function is estimated using a linear combination of a dictionary of pre-specified basis functions. By assuming that the coefficient matrix admits a sparse low-rank structure, we formulate the function estimation problem as a convex program with the trace norm and the ℓ_1-norm regularization. We propose to employ AG and ADMM algorithms to solve the function estimation problem and also develop efficient algorithms for the key components involved in AG and ADMM. We derive a key property of the optimal solution to the convex program; moreover, based on an assumption associated with the basis functions, we establish a performance bound of the proposed function estimation scheme using the composite regularization. Our simulation studies demonstrate the effectiveness and the efficiency of the proposed formulation.

Chapter 7

Conclusion and Future Directions

In this chapter, I summarize the main contributions of this dissertation and discuss some of the future research directions.

7.1 Summary

In this dissertation, I consider the problems of learning multiple tasks simultaneously as well as modeling the task relationship via a shared low-rank structure. The proposed MTL approaches are formulated as the optimization problems of a common generic form, i.e., minimizing the empirical loss over the pre-specified training data with different structured regularizations.

In the first approach - Factor Selection and Coefficient Estimation in Multivariate Linear Regression, I consider to extract a small set of basis factors for capturing the relatedness of multiple related regression functions. This approach is formulated as a multivariate linear regression problem subject to a trace norm constraint.

In the second approach - Learning a Shared Structure from Multiple Tasks, I consider to learn a shared low-dimensional feature mapping from multiple tasks. This approach is formulated as a regularized formulation called iASO, in which the low-rank structure is induced via an orthonormal constraint. Subsequently, iASO is converted into a convex relaxation called rASO, for which a globally optimal solution can be guaranteed. I also derive an interesting theoretical condition based on which rASO can find a globally optimal solution for iASO.

In the third approach - Learning Incoherent Sparse and Low-rank Patterns from Multiple Tasks, I consider to learn incoherent sparse and low-rank patterns from multiple tasks. This approach is formulated as a linear multi-task learning algorithm in which the sparse and low-rank patterns are induced by a sparse regularization term and a low-rank constraint, respectively.

In the fourth approach - Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning, I consider the scenarios where multiple tasks can be divided into a related-task group and an irrelevant-task group. This leads to a robust multi-task learning (RMTL) formulation which learns multiple tasks simultaneously as well as identifies the irrelevant tasks. I also derive a theoretical bound for characterizing the learning performance of RMTL.

In the fifth approach - Learning Multiple Tasks via Sparse Trace Norm Regularization, I consider the problem of estimating multiple predictive functions from a dictionary of basis functions in the nonparametric regression setting. This approach is formulated as a convex program regularized by the trace norm and the ℓ_1-norm simultaneously. Similarly I theoretically establish a performance bound for the proposed function estimation scheme.

For all of the proposed MTL formulations, I develop efficient algorithms for solving the key components involved in the optimization algorithms. I also conduct theoretical analysis for certain MTL approaches such as deriving the globally optimal solution recovery condition and the performance bound. The proposed MTL approaches are applied on two real-world applications for effectiveness demonstration: (1) Automated annotation of the Drosophila gene expression pattern images; (2) Categorization of the Yahoo web pages. Our experimental results demonstrate the efficiency and effectiveness of the proposed algorithms.

7.2 Future Directions

Design of New Low-Rank Regularizations/Constraints In this dissertation, the proposed MTL approaches employ either the trace norm constraint (regularization) or the orthonormal constraint to induce a shared low-dimensional feature mapping (for capturing the task relatedness). Therefore when solving the proposed MTL formulations, single value decomposition (SVD) is in general involved. It is known that SVD leads to expensive computations and may not be practical for the real-world applications involving large scale data sets. One future research direction is to develop new constraints which induce the shared low-dimensional feature mapping while avoiding the expensive SVD computation in the resulting mathematical formulations.

Detection of Complex Cluster Structures in Multiple Tasks In this dissertation, I consider a robust multi-task learning formulation which divides multiple tasks into two groups, i.e., the group of related tasks and the group of irrelevant tasks. However, in reality, the multiple tasks may consist of complex task-groups, for example, a structure of multiple clusters. One future direction is to systematically integrate the clustering algorithms into the MTL models so that the clustering structures among multiple tasks can be automatically detected.

More Efficient Optimization Algorithms In this dissertation, I mainly focus on employing the gradient-type algorithms to solve the proposed MTL formulations. The gradient-type algorithms avoid the computation of second-order information such as the Hessian matrix and they can be applied for the real-world applications involving large scale data sets. In our empirical study, we observe that this type of algorithms converge very fast when the desirable precision (the change of the objective values in two successive iterations) is moderate. However, if the desirable precision is high, the gradient-type algorithms converge slowly. One future direction is to develop optimization algorithms which can efficiently attain high precision in the desirable solution for the proposed MTL formulations.

Learning Negative Task Relatedness Existing MTL algorithms focus on learning multiple tasks simultaneously by modeling the task relatedness or differentiating the related tasks from irrelevant tasks. In essence, the task relationships can be categorized into positive correlation, negative correlation, and task unrelatedness. Clearly ignoring the existence of negative correlation among multiple tasks will affect the generalization performance. One future direction is to develop multi-task learning algorithms in which all three types of tasks relationship are differentiated and utilized appropriately.

Learning Multiple Clustering Problems Currently the approaches of learning multiple tasks via modeling task relatedness has been widely applied for supervised learning settings and the superior performance has been demonstrated both empirically and theoretically compared to learning multiple tasks separately (in supervised learning as well). One future direction is to apply the idea of learning multiple tasks for clustering problems, in which multiple clustering problems are solved simultaneously with their underlying relatedness appropriately modeled.

BIOGRAPHICAL SKETCH

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[3] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, “Multi-task learning for classification with dirichlet process priors,” *Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[4] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[5] J. Baxter, “A model of inductive bias learning,” *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.

[6] K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang, “Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes,” in *UAI*, 2003, pp. 616–623.

[7] K. Yu, V. Tresp, and S. Yu, “A nonparametric hierarchical bayesian framework for information filtering,” in *SIGIR*, 2004, pp. 353–360.

[8] J. Zhang, Z. Ghahramani, and Y. Yang, “Learning multiple related tasks using latent independent component analysis,” in *NIPS*, 2005.

[9] K. Yu, V. Tresp, and A. Schwaighofer, “Learning gaussian processes from multiple tasks,” in *ICML*, 2005.

[10] T. Evgeniou and M. Pontil, “Regularized multi–task learning,” in *KDD*, 2004.

[11] T. Evgeniou, C. A. Micchelli, and M. Pontil, “Learning multiple tasks with kernel methods,” *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[12] N. D. Lawrence and J. C. Platt, “Learning to learn with the informative vector machine,” in *ICML*, 2004.

[13] E. V. Bonilla, K. M. Chai, and C. K. I. Williams, “Multi-task gaussian process prediction,” in *NIPS*, 2007.

[14] S. Thrun and J. O’Sullivan, “Discovering structure in multiple learning tasks: The TC algorithm,” in *ICML*, 1996.

[15] L. Jacob, F. Bach, and J.-P. Vert, “Clustered multi-task learning: A convex formulation,” in *NIPS*, 2008.

[16] R. K. Ando and T. Zhang, “A framework for learning predictive structures from multiple tasks and unlabeled data,” *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[17] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010.

[18] L. Sun, “Multi-label dimension reduction,” *Arizona State University*.

[19] R. K. Ando, “BioCreative II gene mention tagging system at IBM Watson,” in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 2007.

[20] J. Bi, T. Xiong, S. Yu, M. Dundar, and R. B. Rao, “An improved multi-task learning approach with applications in medical diagnosis,” in *ECML*, 2008.

[21] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng, “Multi-task learning for boosting with application to web search ranking,” in *KDD*, 2010.

[22] A. Quattoni, M. Collins, and T. Darrell, “Learning visual representations using images with captions,” in *CVPR*, 2007.

[23] J. Li, Y. Tian, T. Huang, and W. Gao, “Probabilistic multi-task learning for visual saliency estimation in video,” *International Journal of Computer Vision*, vol. 90, no. 2, pp. 150–165, 2010.

[24] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio, “Categorization by learning and combining object parts,” in *NIPS*, 2001.

[25] C. C. Fowlkes, C. L. L. Hendriks, S. V. Keränen, G. H. Weber, O. Rübel, M.-Y. Huang, S. Chatoor, A. H. DePace, L. Simirenko, C. Henriquez, A. Beaton, R. Weiszmann, S. Celniker, B. Hamann, D. W. Knowles, M. D. Biggin, M. B. Eisen, and J. Malik, “A quantitative spatiotemporal atlas of gene expression in the drosophila blastoderm,” *Cell*, vol. 133, no. 2, pp. 364–374, 2008.

[26] E. Lécuyer, H. Yoshida, N. Parthasarathy, C. Alm, T. Babak, T. Cerovina, T. R. Hughes, P. Tomancak, and H. M. Krause, “Global analysis of mRNA localization reveals a prominent role in organizing cellular architecture and function,” *Cell*, vol. 131, no. 1, pp. 174–187, 2007.

[27] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, “Drosophila gene expression pattern annotation using sparse features and term-term interactions,” in *KDD*, 2009.

[28] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] N. Ueda and K. Saito, “Parametric mixture models for multi-labeled text,” in *NIPS*, 2002.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Springer, 2000.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[33] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, “Dimension reduction and coefficient estimation in multivariate linear regression,” *Journal Of The Royal Statistical Society Series B*, vol. 69, no. 3, pp. 329–346, 2007.

[34] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[35] F. R. Bach, D. Heckerman, and E. Horvitz, “Consistency of trace norm minimization,” *Journal of Machine Learning Research*, vol. 9, pp. 1019–1048, 2008.

[36] M. Fazel, H. Hindi, and S. P. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in *ACL*, 2001.

[37] N. Srebro, J. D. M. Rennie, and T. Jaakkola, “Maximum-margin matrix factorization,” in *NIPS*, 2004.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[39] Z. Lu, R. D. C. Monteiro, and M. Yuan, “Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression,” *Submitted to Mathematical Programming*, 2008.

[40] A. Ben-Tal and A. S. Nemirovskiaei, *Lectures on modern convex optimization*. Society for Industrial Mathematics, 2001.

[41] J. F. Sturm, “Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, no. 11-12, pp. 653–625, 1998.

[42] K. Toh, M. Todd, and R. Tutuncu, “Sdpt3: a matlab software package for semidefinite programming,” *Optimization Methods and Software*, 1999.

[43] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[44] A. S. Nemirovskiaei, “Efficient methods in convex programming,” 1994, lecture Notes.

[45] Y. Nesterov, “Introductory lectures on convex programming,” 1998, lecture Notes.

[46] J. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *Submitted*, 2008.

[47] Y. Nesterov, “Smooth minimization of non-smooth functions,” *Mathematical Programming*, 2004.

[48] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, “Efficient projections onto the l1-ball for learning in high dimensions,” in *ICML*, 2008.

[49] G. Gene and V. L. Charles, *Matrix computations*. Johns Hopkins University Press, 1996.

[50] T. Jebara, “Multi-task feature and kernel selection for svms,” in *ICML*, 2004.

[51] G. Obozinski, B. Taskar, and M. I. Jordan, “Multi-task feature selection,” in *Technical report, Department of Statistics, UC Berkeley*, 2006.

[52] A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in *NIPS*, 2006.

[53] ——, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[54] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization,” *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[55] S. Ji and J. Ye, “An accelerated gradient method for trace norm minimization,” in *ICML*, 2009.

[56] T. K. Pong, P. Tseng, S. Ji, and J. Ye, “Trace norm regularization: Reformulations, algorithms, and multi-task learning,” *SIAM Journal on Optimization*, 2009.

[57] A. Nemirovski, *Efficient Methods in Convex Programming*. Lecture Notes, 1995.

[58] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[59] M. L. Overton and R. S. Womersley, “Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrics,” *Mathematical Programming*, vol. 62, no. 1-3, pp. 321–357, 1993.

[60] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[61] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, “A spectral regularization framework for multi-task structure learning,” in *NIPS*, 2007.

[62] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 1996.

[63] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[64] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, “RCV1: A new benchmark collection for text categorization research,” *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[65] A. Shapiro, “Weighted minimum trace factor analysis,” *Psychometrika*, vol. 47, no. 3, pp. 243–264, 1982.

[66] M. Fazel, H. Hindi, and S. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in *ACC*, 2001.

[67] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[68] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis,” *Submitted for publication*, 2009.

[69] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, “Sparse and low-rank matrix decompositions,” in *SYSID*, 2009.

[70] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization,” in *NIPS*, 2009.

[71] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal of Imaging Science*, vol. 2, pp. 183–202, 2009.

[72] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.

[73] G. A. Watson, “Characterization of the subdifferential of some matrix norms,” *Linear Algebra and its Applications*, no. 170, pp. 33–45, 1992.

[74] J. Liu and J. Ye, “Efficient euclidean projections in linear time,” in *ICML*, 2009.

[75] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

[76] A. Martinez and R. Benavente, “The AR face database,” Tech. Rep., 1998.

[77] N. Ueda and K. Saito, “Single-shot detection of multiple categories of text using parametric mixture models,” in *KDD*, 2002.

[78] J. Chen, L. Tang, J. Liu, and J. Ye, “A convex formulation for learning shared structures from multiple tasks,” in *ICML*, 2009.

[79] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *ICML*, 1997.

[80] S. Yu, V. Tresp, and K. Yu, “Robust multi-task learning with t-processes,” in *ICML*, 2007.

[81] Y. Zhang and D.-Y. Yeung, “Multi-task learning using generalized t process,” in *AISTATS*, 2010.

[82] F. Nie, H. Huang, X. Cai, and C. Ding, “Efficient and robust feature selection via joint l21-norms minimization,” in *NIPS*, 2010.

[83] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan., “A dirty model for multi-task learning,” in *NIPS*, 2010.

[84] H. Xu, C. Caramanis, and S. Sanghavi, “Robust pca via outlier pursuit,” in *NIPS*, 2010.

[85] D. Hsu, S. Kakade, and T. Zhang, “Robust matrix decomposition with outliers,” vol. arXiv:1011.1518, 2010.

[86] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient l2,1-norm minimization,” in *UAI*, 2009, pp. 339–348.

[87] S. Ji and J. Ye, “An accelerated gradient method for trace norm minimization,” in *ICML*, 2009, pp. 457–464.

[88] J. Chen, J. Liu, and J. Ye, “Learning incoherent sparse and low-rank patterns from multiple tasks,” in *KDD*, 2010.

[89] J. Liu, J. Chen, and J. Ye, “Large-scale sparse logistic regression,” in *KDD*, 2009.

[90] J. Liu, L. Yuan, and J. Ye, “An efficient algorithm for a class of fused lasso problems,” in *KDD*, 2010.

[91] J.-J. Moreau, “Proximité et dualité dans un espace hilbertien,” *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[92] J.-F. Cai, E. J. Candes, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[93] D. Goldfarb and S. Ma, “Convergence of fixed point continuation algorithms for matrix rank minimization,” *Submitted to Foundations of Computational Mathematics*.

[94] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization,” *SIAM Review*, no. 3, pp. 471–501, 2010.

[95] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, “Simultaneous analysis of lasso and dantzig selector,” *Annals of Statistics*, vol. 37, pp. 1705–1732, 2009.

[96] K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer, “Taking advantage of sparsity in multi-task learning,” in *COLT*, 2008.

[97] D. L. Wallace, “Bounds on normal approximations to student's and the chi-square distributions,” *Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1121–1130, 1959.

[98] F. Bunea, A. B. Tsybakov, and M. H. Wegkamp, “Aggregation and sparsity via ℓ_1 penalized least squares,” in *COLT*, 2006.

[99] J. Huang, T. Zhang, and D. N. Metaxas, “Learning with structured sparsity,” in *ICML*, 2009.

[100] S. Negahban and M. J. Wainwright, “Estimation of (near) low-rank matrices with noise and high-dimensional scaling,” in *ICML*, 2010.

[101] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[102] T. Zhang, “Some sharp performance bounds for least squares regression with l_1 regularization,” *Annals of Statistics*, vol. 37, pp. 2109–2144, 2009.

[103] A. Rohde and A. B. Tsybakov, “Estimation of high-dimensional low rank matrices,” *Preprint available at 0912.5338v2*, 2010.

[104] Y. C. Pati and T. Kailath, “Phase-shifting masks for microlithography: automated design and mask requirements,” *Journal of the Optical Society of America A*, vol. 11, no. 9, pp. 2438–2452, 1994.

[105] K. C. Toh and S. W. Yun, “An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems,” *Pacific Journal of Optimization*, 2009.

[106] F. Bach, “Consistency of trace norm minimization,” *Journal of Machine Learning Research*, vol. 9, pp. 1019–1048, 2008.

[107] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *CoRR*, vol. abs/0805.4471, 2008.

[108] N. Srebro, J. D. M. Rennie, and T. Jaakkola, “Maximum-margin matrix factorization,” in *NIPS*, 2004.

[109] J. D. M. Rennie and N. Srebro, “Fast maximum margin matrix factorization for collaborative prediction,” in *ICML*, 2005.

[110] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *The Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[111] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, “Pathwise coordinate optimization,” *Annals of Statistics*, vol. 1, pp. 302–332, 2007.

[112] E. Candès and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, pp. 4203–4215, 2005.

[113] P. Zhao and B. Yu, “On model selection consistency of lasso,” *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[114] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of ACM*, 2011.

[115] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, 2010.

[116] Y. Nesterov, "Gradient methods for minimizing composite objective function," *CORE Discussion Paper*, 2007.

[117] L. Grippoa and M. Sciandrone, "On the convergence of the block nonlinear gaussĺcseidel method under convex constraints," *Operation Research Letters*, vol. 26, pp. 127–136, 2000.

[118] S. J. Szarek, "Condition numbers of random matrices," *Journal of Complexity*, vol. 7, no. 2, pp. 131–149, 1991.