Statistica Sinica Preprint No: SS-2024-0310

Title	Auxiliary Learning and its Statistical Understanding
Manuscript ID	SS-2024-0310
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202024.0310
Complete List of Authors	Hanchao Yan, Feifei Wang, Chuanxin Xia and Hansheng Wang
Corresponding Authors	Feifei Wang
E-mails	feifei.wang@ruc.edu.cn
Notice: Accepted author version.	

Auxiliary Learning and its Statistical Understanding

Hanchao Yan[1], Feifei Wang[2], Chuanxin Xia[3], and Hansheng Wang[1]

[1] *Peking University;* [2] *Renmin University of China;*

[3] *Ministry of Commerce People's Republic of China.*

Abstract: Modern statistical analysis often encounters high-dimensional problems but with a limited sample size. It poses great challenges to traditional statistical estimation methods. In this work, we adopt auxiliary learning to solve the estimation problem in high-dimensional settings. We start with the linear regression setup. To improve the statistical efficiency of the parameter estimator for the primary task, we consider several auxiliary tasks, which share the same covariates with the primary task. Then a weighted estimator for the primary task is developed, which is a linear combination of the ordinary least squares estimators of both the primary task and auxiliary tasks. The optimal weight is analytically derived and the statistical properties of the corresponding weighted estimator are studied. We then extend the weighted estimator to generalized linear regression models. Extensive numerical experiments are conducted to verify our theoretical results. Last, a deep learning-related real-data example of smart vending machines is presented for illustration purposes.

Key words and phrases: Auxiliary Learning, Deep Learning, High-Dimensional Data Analysis, Ordinary Least Squares Estimator, Smart Vending Machines.

The corresponding author. Email: feifei.wang@ruc.edu.cn

1. INTRODUCTION

Modern statistical analysis frequently encounters datasets characterized by an extensive number of features, which even surpass the sample size. This is attributed to advancements in state-of-the-art technology, which facilitates more convenient collection of data and computing (Gao et al., 2022; Jiang et al., 2023; Li et al., 2024). For instance, in genetic studies, there are hundreds of thousands of single-nucleotide polymorphisms (SNPs) that serve as potential predictors. However, the sample size is limited due to the expensive cost of sequencing. In deep learning-related studies, millions or even billions of features are extracted from texts, images, and other forms of unstructured data. However, the sample size is also relatively limited due to the high cost of labeling. The fact that the feature dimension is larger than the sample size imposes significant challenges to classical statistical analysis. Consider the linear regression model as an example. To estimate the regression coefficient β, the popularly used ordinary least squares (OLS) estimator can achieve excellent estimation performance under the setting of fixed p and $N \to \infty$. However, in the high-dimensional setting with $p \to \infty$ as $N \to \infty$, the finite sample performance of the OLS estimator is no longer satisfactory (Wainwright, 2019). Then how to improve the estimation accuracy becomes a problem of great interest.

To solve this problem, a variety of statistical methods have been developed, which can be roughly classified into two streams. The first stream is penalized estimation, including LASSO (Tibshirani, 1996), the SCAD (Fan and Li, 2001), group LASSO (Yuan and Lin, 2005), adaptive LASSO (Zou, 2006), prior LASSO (Jiang et al., 2016), and many others. The second stream is feature screening. Along this direction, one notable work is the Sure Independence Screening (SIS) method proposed by Fan and Lv (2008). Wang (2009) further improved SIS by applying a forward regression algorithm to screen important features and demonstrated its screening consistency property under an ultrahigh-dimensional setup. The ultrahigh dimensional screening problem but with a low-dimensional factor structure was studied by Wang (2012). Li et al. (2012) proposed a model-free feature screening method based on the distance correlation. Recently, Wang et al. (2021) developed a sequential term selection method for textual datasets, and Zhu et al. (2022) discussed this feature screening method in the context of subsampling. More discussions about the statistical methods in these two streams can be found in Fan et al. (2020).

In this work, we try to address this problem from a different perspective. That is, **auxiliary learning**, which is a popular concept in the field of machine learning (Kung et al., 2021; Li and Dong, 2021; Chen et al., 2022). In auxiliary learning, the research of main interest is often referred to as the primary task. In addition to the primary task, several auxiliary tasks are also considered to assist the primary task in learning better representations or improving the overall performance. To optimize the performance for the primary task, previous researches about auxiliary learning often adopt weighted loss functions to leverage the importance of auxiliary tasks. For example, Liebel and Körner (2018) directly treated the weights of different tasks as parameters to be learned with certain regularization. Navon et al. (2020) adopted a separate network structure to combine all losses into one single objective function, so that nonlinear interactions between different tasks can be learned. Chen et al. (2022) also adopted the approach of using separate network. However, they focused on evaluating the relevance of data features to different tasks, which were then combined to form the weighted loss function.

Note that, auxiliary learning is a special practice of multi-task learning. However, different from auxiliary learning, in multi-task learning, the model is trained on multiple tasks simultaneously to improve the performance of each task by leveraging shared representations (Vandenhende et al., 2021). To achieve representation sharing, a commonly used technique is to use hard-parameter sharing (Kokkinos, 2017; Chen et al., 2018; Kendall et al., 2018) or soft-parameter sharing (Misra et al., 2016; Ruder et al., 2019). Both techniques mainly focus on the shared layers between the tasks to improve the performance of the specific tasks.

Auxiliary learning has been widely adopted in practice (Kanezaki et al., 2018; Howard and Ruder, 2018). Take the real application of smart vending machines in Section 4 as an example. The primary task here is to accurately identify each product of bottled water via the shopping images. It is a multi-class classification problem. Preliminary findings revealed that solely classifying this task yielded suboptimal results. To enhance the classification accuracy, several auxiliary tasks can be considered. For example, we observed that factors such as the cap color, the liquid color, the label shape, whether there are horizontal ridges or vertical ridges contribute to identifying the product of bottled water. Thus identifying these factors can be regarded as auxiliary tasks to enhance the learning process of the primary tasks.

In this work, we aim to study auxiliary learning from a statistical perspective. We start with the linear regression setup for its analytical simplification. However, both our numerical and real data experiments suggest that the theoretical insights obtained here can be readily applied to more sophisticated models (e.g., logistic regression and deep learning). Let $(X_i, Y_i^{(0)})$ be the observation collected from the i-th subject with $1 \leq i \leq N$. Here $Y_i^{(0)}$ is a continuous response of the primary task and $X_i \in \mathbb{R}^p$ is the associated covariate. To model the regression relationship between X_i and $Y_i^{(0)}$, a standard linear regression model can be assumed to be $Y_i^{(0)} = X_i^\top \beta^{(0)} + \varepsilon_i^{(0)}$, where $\beta^{(0)}$ is the primary regression coefficient and $\varepsilon_i^{(0)}$ is the independent error term. Except for $Y_i^{(0)}$, we also consider K auxiliary responses $Y_i^{(k)}$ with $1 \leq k \leq K$. Accordingly, K linear regression models can be established with $Y_i^{(k)} = X_i^\top \beta^{(k)} + \varepsilon_i^{(k)}$. It is notable that, both the primary task and auxiliary tasks share the same covariate vector X_i. Define $\widehat{\beta}_{\text{ols}}^{(k)} \in \mathbb{R}^p$ with $0 \leq k \leq K$ to be the OLS estimator for each task, and let $\widehat{B} = (\widehat{\beta}_{\text{ols}}^{(0)}, \widehat{\beta}_{\text{ols}}^{(1)} ..., \widehat{\beta}_{\text{ols}}^{(K)}) \in \mathbb{R}^{p\times(K+1)}$ be the estimator matrix for all tasks. In order to borrow information from the auxiliary tasks, we develop a weighted estimator for $\beta^{(0)}$ as $\widehat{\beta}_w = \widehat{B}w$, where $w \in \mathbb{R}^{K+1}$ is the associated weight vector. The optimal weight $\widehat{w}$ can be computed, which leads to the feasible weighted estimator $\widehat{\beta}_{\widehat{w}}$. We empirically and theoretically prove that the weighted estimator can enjoy good statistical efficiency. Finally, we apply the weighted estimator to solve the product classification problem of smart vending machines.

The rest of this article is organized as follows. Section 2 presents the estimation method for the linear regression model with auxiliary learning. The theoretical properties of the proposed weighted estimator are also rigorously studied. Section 3 presents extensive numerical experiments to demonstrate the finite sample performance of the weighted estimator. We also extend the weighted estimator to generalized linear regression models and demonstrate its performance empirically in this section. Section 4 is the real application of smart vending machines. By using the weighted estimator, the classification accuracy can be largely improved. Section 5 concludes the paper with a brief discussion.

2. THE ESTIMATION THEORY

2.1 The Model Setup

Let $(X_i, Y_i^{(0)})$ be the observation collected from the i-th subject with $1 \leq i \leq N$. Here $Y_i^{(0)} \in \mathbb{R}$ is the continuous response of primary interest and $X_i = (X_{i1}, ..., X_{ip})^\top \in \mathbb{R}^p$ is the associated p-dimensional feature vector with covariance matrix $E(X_i X_i^\top) = \Sigma_{xx} \in \mathbb{R}^{p\times p}$. To model their regression relationship, a standard linear regression model can be assumed as follows:

$$Y_i^{(0)} = X_i^\top \beta^{(0)} + \varepsilon_i^{(0)}, \tag{2.1}$$

where $\beta^{(0)} = (\beta_1^{(0)}, ..., \beta_p^{(0)})^\top \in \mathbb{R}^p$ is the associated regression coefficient vector and $\varepsilon_i^{(0)} \in \mathbb{R}$ is the associated random noise with $E(\varepsilon_i^{(0)}) = 0$ and $\mathrm{var}(\varepsilon_i^{(0)}) = \sigma_0^2$. To estimate the regression coefficient $\beta^{(0)}$, an ordinary least squares type objective function can be constructed as $\mathcal{L}^{(0)}(\beta) = N^{-1}\sum_{i=1}^N (Y_i^{(0)} - X_i^\top\beta)^2$. Let $\widehat{\Sigma}_{xx} = N^{-1}\sum_{i=1}^N X_i X_i^\top$ and $\widehat{\Sigma}_{xy}^{(0)} = N^{-1}\sum_{i=1}^N Y_i^{(0)} X_i$. Then, an ordinary least squares (OLS) estimator can be obtained as $\widehat{\beta}_{\text{ols}}^{(0)} = \arg\min_{\beta\in\mathbb{R}^p} \mathcal{L}^{(0)}(\beta) = \widehat{\Sigma}_{xx}^{-1}\widehat{\Sigma}_{xy}^{(0)}$. Standard asymptotic theory suggests that $E(\widehat{\beta}_{\text{ols}}^{(0)}) = \beta^{(0)}$ and $N^{-1/2}\widehat{\Sigma}_{xx}^{1/2}(\widehat{\beta}_{\text{ols}}^{(0)} - \beta^{(0)}) \to_d \mathcal{N}(0, \sigma_0^2 I_p)$, where I_p stands for a p-dimensional identity matrix (Van der Vaart, 2000; Murray, 2012). Furthermore, the finite sample performance of $\widehat{\beta}_{\text{ols}}^{(0)}$ is also widely demonstrated in the past literature. Specifically, it can be verified that the estimation accuracy of $\widehat{\beta}_{\text{ols}}^{(0)}$ measured by $\|\widehat{\beta}_{\text{ols}} - \beta^{(0)}\|_2$ is of the order $O_p(\sqrt{p/N})$ (Wainwright, 2019).

Assume we can also observe K continuous variables as $Y_i^{(k)} \in \mathbb{R}$ with $1 \leq k \leq K$ for the i-th subject. These additional variables might be closely correlated to the primary response $Y_i^{(0)}$. Then we are able to adopt the idea of auxiliary learning (AL) to improve the estimation accuracy of $\beta^{(0)}$. Auxiliary learning is a machine learning technique, where a model is trained on a primary task, along with one or more auxiliary tasks that are related but not the main focus of interest. The objective of these auxiliary tasks is to provide additional guidance or regularization to the model, thus improving the estimation performance on the primary task. In auxiliary learning, K is the total number of auxiliary tasks, and $Y_i^{(k)}$ with $1 \le k \le K$ are referred to as the auxiliary responses. Assume for each auxiliary response $Y_i^{(k)}$, there exists a linear regression model as

$$Y_i^{(k)} = X_i^\top \beta^{(k)} + \varepsilon_i^{(k)}, \tag{2.2}$$

where $\beta^{(k)} \in \mathbb{R}^p$ is the associated regression coefficient and $\varepsilon_i^{(k)} \in \mathbb{R}$ is the random noise of the i-th subject with $E(\varepsilon_i^{(k)}) = 0$ and $\mathrm{var}(\varepsilon_i^{(k)}) = \sigma_k^2$. Let $\varepsilon_i = (\varepsilon_i^{(0)}, ..., \varepsilon_i^{(K)})^\top \in \mathbb{R}^{K+1}$ be the collection of random noises for the i-th subject from all tasks. Assume $E(\varepsilon_i) = \mathbf{0}_{K+1}$ and $\mathrm{cov}(\varepsilon_i) = \Sigma_\varepsilon$ for some covariance matrix $\Sigma_\varepsilon = (\sigma_{k_1,k_2}) \in \mathbb{R}^{(K+1)\times(K+1)}$. We next denote $B = (\beta^{(0)}, ..., \beta^{(K)}) \in \mathbb{R}^{p\times(K+1)}$ to collect the true regression coefficients for both primary task and auxiliary tasks. For each $\beta^{(k)}$ with $1 \le k \le K$, we can similarly obtain an OLS estimator as $\widehat{\beta}_{\text{ols}}^{(k)} = \widehat{\Sigma}_{xx}^{-1}\widehat{\Sigma}_{xy}^{(k)}$, where $\widehat{\Sigma}_{xy}^{(k)} = N^{-1}\sum_{i=1}^N Y_i^{(k)} X_i$. This leads to the estimated coefficient matrix $\widehat{B} = (\widehat{\beta}_{\text{ols}}^{(0)}, ..., \widehat{\beta}_{\text{ols}}^{(K)}) \in \mathbb{R}^{p\times(K+1)}$. Meanwhile, we can obtain the estimator of the covariance matrix as $\widehat{\Sigma}_\varepsilon = (\widehat{\sigma}_{k_1,k_2}) \in \mathbb{R}^{(K+1)\times(K+1)}$, where $\widehat{\sigma}_{k_1,k_2} = N^{-1}\sum_{i=1}^N (Y_i^{(k_1)} - X_i^\top \widehat{\beta}_{\text{ols}}^{(k_1)})(Y_i^{(k_2)} - X_i^\top \widehat{\beta}_{\text{ols}}^{(k_2)})$. Then the key question is how to exploit the information of $\widehat{B}$ and $\widehat{\Sigma}_\varepsilon$ to improve the estimation performance of $\beta^{(0)}$.

2.2 A Weighted Estimator

To take advantage of the information provided by the auxiliary tasks, we study here a weighted estimator. Specifically, we assume the true coefficient matrix B is of rank $d < K+1$, which potentially guarantees that $\beta^{(0)}$ can be well approximated by the linear combination of $\beta^{(k)}$s. More specifically, consider a weight vector $w = (w_0, ..., w_K)^\top \in \mathbb{R}^{K+1}$. Then let $\widehat{\beta}_w = \widehat{B}w$ be a weighed estimator. Since $\widehat{B}$ contains the OLS estimators for B, we should have $E(\widehat{B}) = B$ and thus $E(\widehat{\beta}_w) = Bw$. Recall that the ultimate goal here is to estimate the primary parameter $\beta^{(0)}$. Consequently, our goal here is to find a weight w satisfying the constraint $E(\widehat{\beta}_w) = Bw = \beta^{(0)}$.

Let $\mathbb{W} = \{w \in \mathbb{R}^{K+1} : Bw = \beta^{(0)}\}$ be a feasible solution set. We then discuss the property of $\mathbb{W}$. First, we know immediately that $\mathbb{W}$ is not empty, since the vector $e_1 = (1, 0, \cdots, 0)^\top \in \mathbb{R}^{K+1}$ is contained in $\mathbb{W}$. Second, for any $w \in \mathbb{W}$, we should have $B(w - e_1) = \mathbf{0}_p$. Note that B is of rank $d < K + 1$. Thus $\mathbb{W}$ should have infinite choices of w. Then how to find an appropriate choice of w from $\mathbb{W}$ is an important issue. To this end, we conduct the following two steps. In the first step, we provide an analytic form of $\mathbb{W}$; while in the second step, under this form of $\mathbb{W}$, we find an optimal choice of w.

Step 1. Consider the eigenvalue decomposition for $B^\top B$ as $B^\top B = \sum_{k=1}^{K+1} \lambda_k \nu_k \nu_k^\top$, where λ_k is the k-th largest eigenvalue and $\nu_k = (\nu_{k1}, ..., \nu_{kK+1})^\top$
$\in \mathbb{R}^{K+1}$ is the associated eigenvector. Write $j_{\max}^{(k)} = \arg\max_j\{|\nu_{kj}| : 1 \leq j \leq K + 1\}$. By requiring $\nu_{kj_{\max}^{(k)}} > 0$, both the λ_ks and ν_ks can be uniquely identified. Recall that $\mathrm{rank}(B) = d$ and thus $\mathrm{rank}(B^\top B) = d$. We then have $\lambda_k = 0$ for any $d < k \leq K + 1$. Define $\Theta = (\nu_{d+1}, ..., \nu_{K+1}) \in \mathbb{R}^{(K+1)\times(K-d+1)}$. By simple calculations, we have $\Theta^\top \Theta = I_{K+1-d}$ and $\Theta^\top B^\top B \Theta = \boldsymbol{O}_{K-d+1}$. Then with the help of Θ, we can obtain an analytical form of $\mathbb{W}$ as $\mathbb{W} = \{e_1 + \Theta u : u \in \mathbb{R}^{K-d+1}\}$. The detailed derivations are given in Appendix A.1 in the supplementary materials.

Step 2. We next consider how to select an optimal weight $w \in \mathbb{W}$. Note for any

$w \in \mathbb{W}$, we have $E(\widehat{\beta}_w) = E(\widehat{B}w) = \beta^{(0)}$ and the covariance $\text{cov}(\widehat{\beta}_w) = N^{-1}w^\top \Sigma_\varepsilon w E(\widehat{\Sigma}_{xx}^{-1})$. This suggests that $\widehat{\beta}_w$s share the same mean $\beta^{(0)}$ but with different covariance. Thus we are motivated to select an optimal weight $w^* \in \mathbb{W}$ that minimizes the covariance, thereby achieving the highest statistical efficiency. In other words, w^* should satisfy the condition that $\text{cov}(\widehat{\beta}_w) - \text{cov}(\widehat{\beta}_{w^*})$ is semi-positive definite for all $w \in \mathbb{W}$. Further define $\mathcal{P}(w) = w^\top \Sigma_\varepsilon w \in \mathbb{R}$, which leads to $\text{cov}(\widehat{\beta}_w) = N^{-1}\mathcal{P}(w)E(\widehat{\Sigma}_{xx}^{-1})$. Note that the choice for weight w only affects the term $\mathcal{P}(w)$ and has nothing to do with the other terms in $\text{cov}(\widehat{\beta}_w)$. Therefore, the optimal choice about w^* is given by $w^* = \arg\min_{w\in\mathbb{W}} \mathcal{P}(w)$. By simple calculations, we can derive

$$w^* = e_1 - \Theta(\Theta^\top \Sigma_\varepsilon \Theta)^{-1}\Theta^\top \Sigma_\varepsilon e_1. \tag{2.3}$$

More details about the derivation of w^* can be found in Appendix A.2 in the supplementary materials.

Remark 1. By (2.3), we find that the optimal weight w^* is affected by the covariance Σ_ε. In particular, the noise level of each task (i.e., σ_k^2) plays an important role in determining w^*. To see this, consider for example a special case with $\Sigma_\varepsilon = \text{diag}(\sigma_k^2) \in \mathbb{R}^{(K+1)\times(K+1)}$ and $\beta^{(k)} = \beta^{(0)}$ for $1 \leq k \leq K$. For this case, we have $w_k^* = \left\{\sigma_k^2(\sum_{k'=0}^K \sigma_{k'}^{-2})\right\}^{-1}$. For an arbitrary task k, we should have $w_k^* \to 0$ if $\sigma_k^2 \to \infty$. This suggests that the optimal weights assigned to tasks with high noise levels should be small. In contrast, we should have $w_k^* \to 1$ if $\sigma_k^2 \to 0$. This suggests that the optimal weights assigned to tasks with low noise levels should be large.

Remark 2. We also find that the optimal weight w^* is affected by the low-rank structure of B through Θ. To see this, consider for example the case where B is of full rank structure with $d = \text{rank}(B) = K + 1$. In this case, Θ becomes infeasible. Recall that for any $w \in \mathbb{W}$, we must have $Bw = \beta^{(0)}$ and $Be_1 = \beta^{(0)}$. It follows that $B(w - e_1) = \mathbf{0}_p$. Since B is of full rank, we must have $w - e_1 = \mathbf{0}_{K+1}$. This suggests that e_1 is the only vector contained in the feasible set $\mathbb{W}$. Therefore, we must have $w^* = e_1$. In other words, the weights for all auxiliary tasks should be zero, i.e., $w_k^* = 0$ for $1 \leq k \leq K$. Therefore all the auxiliary tasks are useless and should be excluded.

By plugging in the optimal weight w^* into $\widehat{\beta}_w$, we obtain the estimator $\widehat{\beta}_{w^*} = \widehat{B}w^*$, which is referred to as the "ORACLE" estimator. However, the ORACLE estimator cannot be used immediately, since the optimal weight w^* involves two unknown parameters (i.e., Θ and Σ_ε). To find a practically feasible estimator, both Θ and Σ_ε need to be estimated. In Section 2.1, we have already obtained an estimator for the covariance matrix as $\widehat{\Sigma}_\varepsilon$. Therefore the key problem here is to find an estimator for Θ. To this end, we seek the help from the estimated coefficient matrix $\widehat{B}$. Specifically, we can conduct the eigenvalue decomposition on $\widehat{B}^\top \widehat{B}$. Let $\widehat{\lambda}_k$ be the k-th largest eigenvalue and $\widehat{\nu}_k$ be the associated eigenvector. Therefore, w^* can be estimated as $\widehat{w}^* = e_1 - \widehat{\Theta}(\widehat{\Theta}^\top \widehat{\Sigma}_\varepsilon \widehat{\Theta})^{-1} \widehat{\Theta}^\top \widehat{\Sigma}_\varepsilon e_1$, where $\widehat{\Theta} = (\widehat{\nu}_{d+1}, ..., \widehat{\nu}_{K+1}) \in \mathbb{R}^{(K+1)\times(K-d+1)}$. We then plug $\widehat{w}^*$ into $\widehat{\beta}_{w^*}$ and this leads to the final weighted estimator $\widehat{\beta}_{\widehat{w}^*}$, which is referred to as the "FEASIBLE" estimator.

2.3 Convergence Rate

In this section, we study the theoretical properties of the feasible estimator $\widehat{\beta}_{\widehat{w}^*}$. To facilitate the theoretical analysis, we first introduce some useful notations. For any arbitrary matrix A, define the operator norm as $\|A\|_{\text{op}} = \lambda_{\max}^{1/2}(A^\top A)$, where $\lambda_{\max}(M)$ stands for the maximum absolute eigenvalue of an arbitrary symmetric matrix M. Furthermore, denote the Frobenius norm for A as $\|A\|_F = \text{tr}^{1/2}(A^\top A)$. Moreover, for any arbitrary vector u, define its ℓ_2-norm as $\|u\|_2 = (u^\top u)^{1/2}$. Next, for a random variable $U \in \mathbb{R}$, define its sub-Gaussian norm as $\|U\|_{\psi_2} = \inf\{t > 0 : E\{\exp(U^2/t^2)\} \leq 2\}$. For a random vector $V \in \mathbb{R}^p$, define its sub-Gaussian norm as $\|V\|_{\psi_2} = \sup_{\|u\|=1} \|u^\top V\|_{\psi_2}$. We then say a random vector V is a sub-Gaussian random variable if $\|V\|_{\psi_2} < \infty$. Please see Vershynin (2018) and Wainwright (2019) for more discussions about the theoretical properties of sub-Gaussian variables. To study the non-asymptotic behavior of the proposed FEASIBLE estimator, some commonly used technical conditions are needed and given as follows.

(C1) (Sub-Gaussian Condition) Assume $V_i = (X_i^\top, \varepsilon_i^\top)^\top \in \mathbb{R}^{p+K+1}$ is an independent and identically distributed sub-Gaussian random vector with $\|V_i\|_{\psi_2} = C_{\text{sub}} < \infty$.

(C2) (Variance Condition) Assume there are two constants $\tau_{\min}$ and $\tau_{\max}$ such that (1) $0 < \tau_{\min} \leq \lambda_{\min}(\Sigma_{xx}) \leq \lambda_{\max}(\Sigma_{xx}) \leq \tau_{\max} < \infty$ as $p \to \infty$; and (2) $0 < \tau_{\min} \leq \lambda_{\min}(\Sigma_\varepsilon) \leq \lambda_{\max}(\Sigma_\varepsilon) \leq \tau_{\max} < \infty$ as $K \to \infty$.

(C3) (Regression Coefficients) Assume $\text{rank}(B) = d < K+1$. For the same constants $0 < \tau_{\min} \leq \tau_{\max} < \infty$ as in Condition (C2), we assume that $\tau_{\min} \leq \lambda_d(B^\top B) \leq \lambda_1(B^\top B) = \lambda_{\max}(B^\top B) \leq \tau_{\max}$ as $p \to \infty$. Here $\lambda_j(A)$ stands for the j-th largest eigenvalue for an arbitrary symmetric matrix A.

(C4) (Divergence Rate) Assume that (1) $K \to \infty$, (2) $p \to \infty$, (3) $K/p \to 0$, and (4) $p/N \to 0$ as $N \to \infty$.

Condition (C1) is a standard sub-Gaussian assumption commonly used in high-dimensional researches (Vershynin, 2018; Wainwright, 2019; Zhu et al., 2022; Tian and Feng, 2023). Condition (C2) imposes a fixed lower bound for $\lambda_{\min}(\Sigma_{xx})$ so that different features are not allowed to be linearly dependent with each other severely. In the meanwhile, Condition

(C2) imposes a fixed upper bound on $\lambda_{\max}(\Sigma_{xx})$ so that the variance of X_i is bounded. The second condition in (C2) is used to prevent the random noises of different tasks from being seriously correlated. Conditions of similar type have been widely used in the past literature (Fan et al., 2008; Zhu et al., 2020; Fan et al., 2024). Condition (C3) regularizes the low-rank structure for the regression coefficient B. On one side, it provides an upper bound for $\lambda_{\max}(B^\top B)$, so that the information contained in $X_i\beta^{(k)}$s would not explode to infinity. On the other side, the lower bound of $\lambda_d(B^\top B)$ enforces that different $\beta^{(k)}$s should not be linearly dependent with each other severely. Condition (C4) specifies the divergence rates of K, p, and N. Under the above conditions, we have the following theorem.

Theorem 1. *Assume the technical conditions (C1)—(C4) hold. Then there exist constants C_1 and C_2, such that the following inequality holds with probability at least $1 - C_2\exp(-K)$, as long as $N > N_0$ for some sufficiently large constant N_0*

$$\left\|\widehat{\beta}_{\widehat{w}^*} - \widehat{\beta}_{w^*}\right\|_2 \leq C_1\left\{\sqrt{\frac{K+d}{N}} + \frac{p}{N}\right\}. \tag{2.4}$$

The detailed proof of Theorem 1 is given in Appendix B in the supplementary materials. Theorem 1 provides an upper bound for the discrepancy between the FEASIBLE estimator $\widehat{\beta}_{\widehat{w}^*}$ and the ORACLE estimator $\widehat{\beta}_{w^*}$. From Theorem 1, we know immediately that $\|\widehat{\beta}_{\widehat{w}^*} - \widehat{\beta}_{w^*}\|_2 = O_p\{\sqrt{(K+d)/N} + p/N\}$ as $N \to \infty$. This finding suggests that the discrepancy between $\widehat{\beta}_{\widehat{w}^*}$ and $\widehat{\beta}_{w^*}$ goes to zero as $N \to \infty$.

Based on Theorem 1, we further study the discrepancy between the FEASIBLE estimator $\widehat{\beta}_{\widehat{w}^*}$ and the true parameter $\beta^{(0)}$. To this end, we first study the theoretical behaviour of the ORACLE estimator $\widehat{\beta}_{w^*}$. Note that $\widehat{\beta}_{w^*}$ is an unbiased estimator for $\beta^{(0)}$, i.e., $E(\widehat{\beta}_{w^*}) = \beta^{(0)}$.

Moreover, since $\widehat{\beta}_{w^*}$ is a linear combination of several OLS estimators, it can be verified that $\sqrt{N}\alpha^\top(\widehat{\beta}_{w^*} - \beta^{(0)}) \rightarrow_d \mathcal{N}\Big(0, \mathcal{P}(w^*)\alpha^\top\Sigma_{xx}^{-1}\alpha\Big)$, where $\alpha = (\alpha_1, ..., \alpha_p)^\top \in \mathbb{R}^p$ is an arbitrary p-dimensional, non-zero, and fixed vector. This suggests that, the difference between $\widehat{\beta}_{w^*}$ and $\beta^{(0)}$ is of the order $O_p(\sqrt{p/N})$, in the sense that $\alpha^\top(\widehat{\beta}_{w^*} - \beta^{(0)}) = O_p(\sqrt{p/N})$. On the other side, Theorem 1 suggests that $\alpha^\top(\widehat{\beta}_{\widehat{w}^*} - \widehat{\beta}_{w^*}) = O_p\{\sqrt{(K+d)/N} + p/N\}$ as $N \rightarrow \infty$. It is a small order term as compared with $\alpha^\top(\widehat{\beta}_{w^*} - \beta^{(0)}) = O_p(\sqrt{p/N})$ since $d < K + 1$, $K/p \rightarrow 0$, and $p/N \rightarrow 0$ as $N \rightarrow \infty$. Therefore, the difference between $\widehat{\beta}_{\widehat{w}^*}$ and $\widehat{\beta}_{w^*}$ can be asymptotically ignorable. This suggests that $\alpha^\top(\widehat{\beta}_{\widehat{w}^*} - \beta^{(0)}) = \alpha^\top(\widehat{\beta}_{\widehat{w}^*} - \widehat{\beta}_{w^*}) + \alpha^\top(\widehat{\beta}_{w^*} - \beta^{(0)})$ should have the same asymptotic distribution as $\alpha^\top(\widehat{\beta}_{w^*} - \beta^{(0)})$. Therefore, we have the following corollary.

Corollary 1. *Assume the technical conditions (C1)—(C4) hold. Then as $N \rightarrow \infty$, we have* $\sqrt{N}\alpha^\top(\widehat{\beta}_{\widehat{w}^*} - \beta^{(0)}) \rightarrow_d \mathcal{N}\Big(0, \mathcal{P}(w^*)\alpha^\top\Sigma_{xx}^{-1}\alpha\Big)$.

3. SIMULATION STUDY

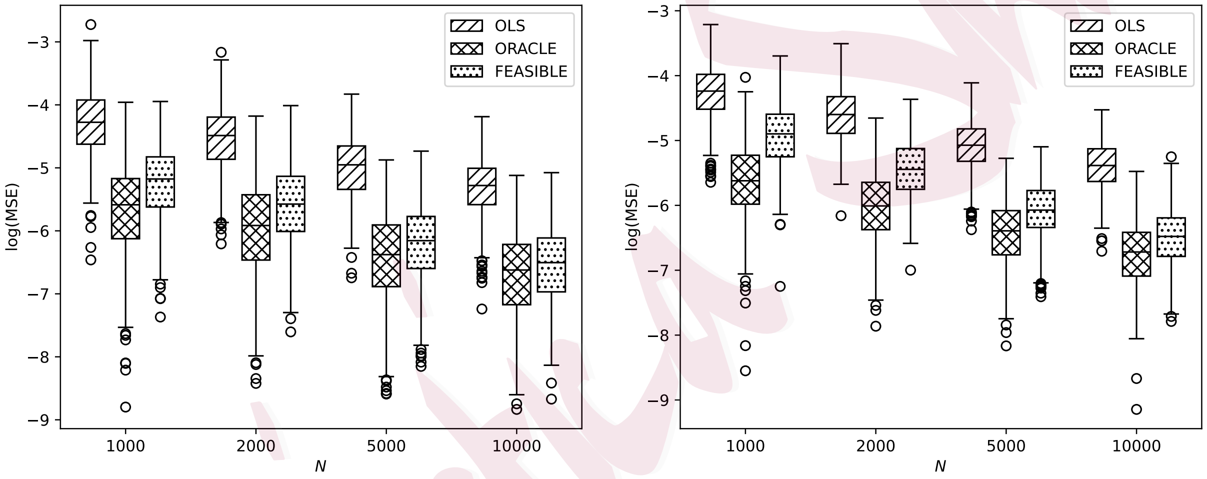
3.1 Simulation Setup and Performance Measure

To demonstrate the finite sample performance of the weighted estimators, we conduct a number of simulation experiments in this section. We start with the primary linear regression model (2.1). The feature vector X_i with $1 \leq i \leq N$ is generated from a multivariate Gaussian distribution with mean $\mathbf{0}_p$ and covariance matrix $\Sigma_{xx} = (\sigma_{j_1,j_2}) \in \mathbb{R}^{p\times p}$, where $\sigma_{j_1,j_2} = 0.5^{|j_1-j_2|}$ for $1 \leq j_1, j_2 \leq p$ (Tibshirani, 1996; Fan and Lv, 2008). Recall that $B \in \mathbb{R}^{p\times(K+1)}$ is the true coefficient matrix with rank d. We generate B as $B = \Phi_1\Phi_2/\|\Phi_1\Phi_2\|_{\text{op}}$ with $\Phi_1 \in \mathbb{R}^{p\times d}$ and $\Phi_2 \in \mathbb{R}^{d\times(K+1)}$. Here $\Phi_1 = (\phi_{ij}^{(1)})$ with $\phi_{ij}^{(1)}$ independently generated from $\mathcal{N}(0, 1/\sqrt{p})$; $\Phi_2 = (\phi_{ij}^{(2)})$, where $\phi_{ij}^{(2)} = 0$ if $i \leq j \leq i + K - d + 1$ and $\phi_{ij}^{(2)} = 1$ otherwise. It follows that $\mathrm{rank}(B) = d$ and $\|B\|_{\mathrm{op}} = 1$. The random noise vector $\varepsilon_i = (\varepsilon_i^{(0)}, ..., \varepsilon_i^{(K)})^\top \in \mathbb{R}^{K+1}$ is generated independently from a multivariate normal distribution with mean $\mathbf{0}_{K+1}$ and covariance matrix $\Sigma_\varepsilon \in \mathbb{R}^{(K+1)\times(K+1)}$. Here $\Sigma_\varepsilon = \widetilde{\Sigma}_\varepsilon \widetilde{\Sigma}_\varepsilon^\top / \|\widetilde{\Sigma}_\varepsilon\|_{\mathrm{op}}^2$ and each component of $\widetilde{\Sigma}_\varepsilon \in \mathbb{R}^{(K+1)\times(K+1)}$ is independently generated from $\mathcal{N}(0, 1/\sqrt{K+1})$. It follows that $\|\Sigma_\varepsilon\|_{\mathrm{op}} = 1$. Based on the feature vector X_i, the coefficient matrix B, and the random noise ε_i, the responses $Y_i^{(k)}$ with $0 \le k \le K$ can be generated accordingly.

We randomly replicate the data generation process for a total of $M = 500$ times. For each generated dataset, we compute three estimators for the primary coefficient $\beta^{(0)}$. The first one is the OLS estimator by using $Y_i^{(0)}$ and X_i only. We denote this estimator by "OLS". The second one is the ORACLE estimator $\widehat{\beta}_{w^*}$ using the optimal weight w^*. The last one is the FEASIBLE estimator $\widehat{\beta}_{\widehat{w}^*}$ using the estimated weight $\widehat{w}^*$. For each estimator (i.e., OLS, ORACLE, and FEASIBLE), we define $\widehat{\beta}_{(m)}^{(0)} \in \mathbb{R}^p$ as the estimator for $\beta^{(0)}$ in the mth replication ($1 \le m \le M$). Then, to evaluate the estimation performance of each estimator, we use the mean squared error (MSE), namely, $M^{-1}\sum_{m=1}^{M} \|\widehat{\beta}_{(m)}^{(0)} - \beta^{(0)}\|_2^2$, where $\|\cdot\|_2$ is the ℓ_2-norm.

3.2 Example 1: Varying N and p

We first study the estimation performance of different estimators with varying sample size N and feature dimension p. Specifically, we assume the total sample size $N \in \{1000, 2000, 5000, 10000\}$. Let the feature dimension $p = \sqrt{N}$, so that the feature dimension p diverges as the sample size N increases. Then for each (N, p)-pair, fix the number of auxiliary tasks $K \in \{10, 20\}$ and let $d = \mathrm{rank}(B) = K/2$ respectively. The detailed simulation results are shown in Figure 1, from which we can draw the following conclusions. First, when the sample size (i.e., N) increases, the MSE values of the three estimators steadily decrease. This suggests the consistency of these estimators. Second, regardless of the sample size, the MSE values of the ORACLE and FEASIBLE estimators are consistently smaller than those of the OLS estimator. This finding demonstrates the advantage of the weighted estimators over the OLS estimator with the help of auxiliary tasks. Third, when N is relatively large, the FEASIBLE estimator can achieve almost the same estimation performance as the ORACLE estimator, which is consistent with Theorem 1.

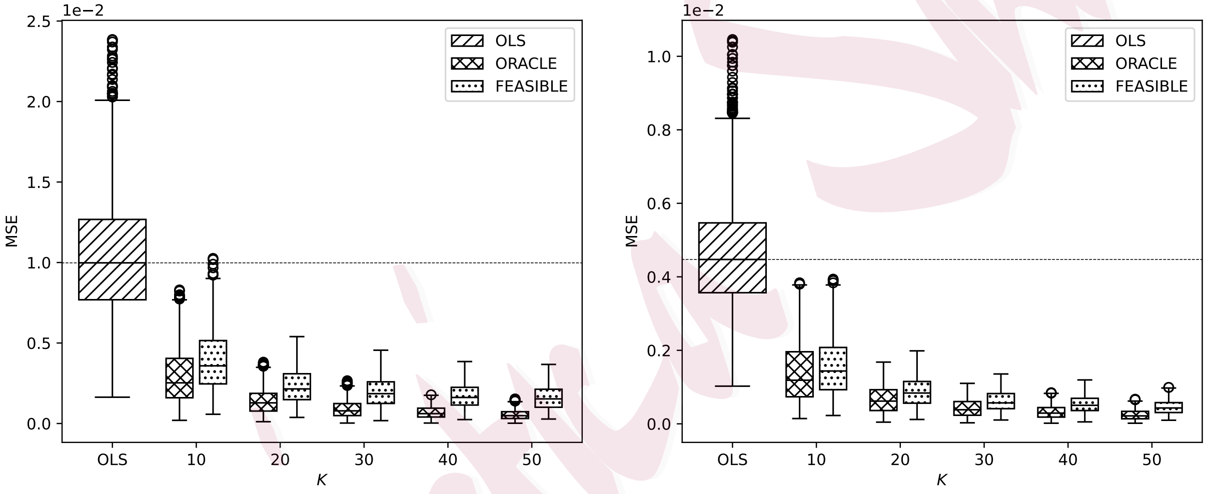


(a) $p = \sqrt{N}, K = 10, d = 5$ (b) $p = \sqrt{N}, K = 20, d = 10$

Figure 1: Boxplots of log-transformed MSE values of the OLS, ORACLE, and FEASIBLE estimators with varying N and p.

3.3 Example 2: Varying K with Fixed d

We next investigate how the number of auxiliary tasks affects the estimation performance of the weighted estimators. In this scenario, we assume that $d = \text{rank}(B)$ remains fixed. In other words, as the number of auxiliary tasks K increases, the newly added coefficient vectors remain within the original column space of B. In this case, the auxiliary tasks are highly correlated with each other. It would lead to a more flexible feasible solution set $\mathbb{W}$, which further yields a smaller $\mathcal{P}(w^*) = \min_{w\in\mathbb{W}} w^\top \Sigma_\varepsilon w$. Then by our Corollary 1, we should expect both ORACLE and FEASIBLE estimators have lower MSE values as K increases. To illustrate this idea, assume $N \in \{2000, 10000\}$ and $p = \sqrt{N}$. Let K vary from 10 to 50 with a step size of 10. For each K, we fix $d = 5$. The corresponding simulation results are shown in Figure 2. We can find that, with a fixed d, the MSE values of both the ORACLE and FEASIBLE estimators decrease as K grows. This finding also verifies that under a fixed d, including more auxiliary tasks should improve the estimation performance.



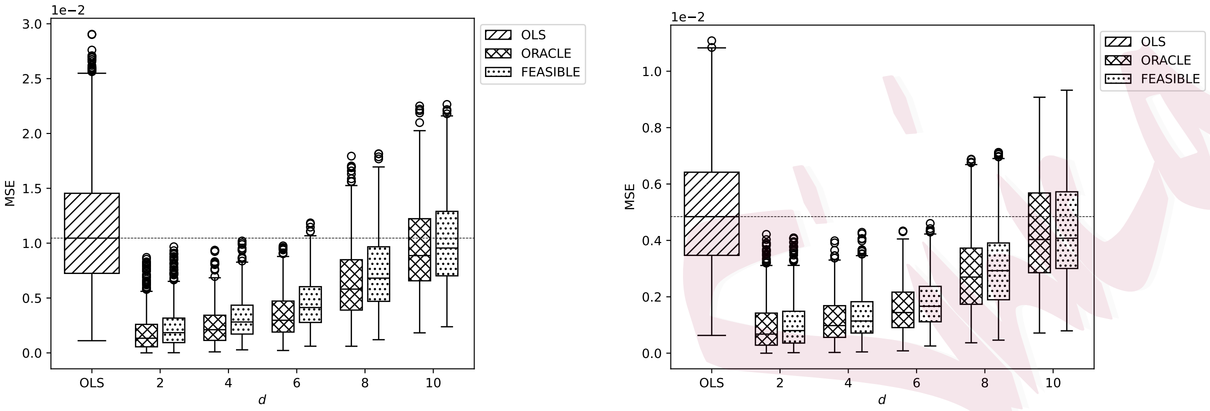
(a) $N = 2000, p = \sqrt{N}, d = 5$

(b) $N = 10000, p = \sqrt{N}, d = 5$

Figure 2: Boxplots of MSE values of the OLS, ORACLE, and FEASIBLE estimators with fixed d but varying K.

3.4 Example 3: Varying d with Fixed K

In this example, we study the effect of the rank d under a fixed number of auxiliary tasks K. With a fixed K, a larger rank d suggests $\beta^{(k)}$s are less likely to be linearly dependent with each other, and thus might introduce extra noise and harm the estimation performance. To verify this idea, we assume $N \in \{2000, 10000\}$, $p = \sqrt{N}$, and fix $K = 10$. Then we investigate the estimation performance of the weighted estimators with varying d. Specifically, let d vary from 2 to 10 with a step size of 2. The corresponding results are shown in Figure 3. By Figure 3, it can be verified that with an increasing d, the MSE values for both ORACLE and FEASIBLE estimators are increasing steadily, which is consistent with our expectation.



(a) $N = 2000, p = \sqrt{N}, K = 10$

(b) $N = 10000, p = \sqrt{N}, K = 10$

Figure 3: Boxplots of MSE values of the OLS, ORACLE, and FEASIBLE estimators with fixed K but varying d.

3.5 Example 4: Estimating with Low-Quality Auxiliary Tasks

In this example, we focus on how the quality of auxiliary tasks affects the estimation performance. Assume we already have a total of K auxiliary tasks, whose coefficient matrix is denoted by B_1 and $\text{rank}(B_1) = d$. Assume all these auxiliary tasks are beneficial to the primary task, with each having an optimal weight $w_k^* \neq 0$ for every $1 \leq k \leq K$. We next introduce an additional set of K' auxiliary tasks, whose coefficient matrix is denoted by $B_2 = (\beta^{(K+1)}, ..., \beta^{(K+K')}) \in \mathbb{R}^{p \times K'}$. Further assume these newly introduced auxiliary tasks are of low-quality relative to the primary task, i.e., their optimal weights are given by $w_k^* = 0$ for $K < k \leq K + K'$. In summary, we have a total of $K + K'$ auxiliary tasks, where the first K tasks are useful and the remaining K' tasks are useless. Next, we fix $N = 10,000$, $p = \sqrt{N} = 100$, $K = 10$, $d = 5$, and $K' = 50$. We then generate the coefficient matrix as follows. Similar to the generation process in Section 3.1, we first separately generate two coefficient matrices $B_1 \in \mathbb{R}^{p\times(K+1)}$ with $\mathrm{rank}(B_1) = d$ and $B_2 \in \mathbb{R}^{p\times K'}$ with $\mathrm{rank}(B_2) = K'$. Then we define $B_2^* = \{I - B_1(B_1^\top B_1)^{-1}B_1^\top\}B_2$, so that B_2^* is linearly independent of B_1. This leads to the final coefficient matrix $B = (B_1, B_2^*) \in \mathbb{R}^{p\times(K+K'+1)}$. Subsequently, the covariate X_i and random noise ε_i are generated similar to Section 3.1, and the response Y_i can also be generated accordingly.

To evaluate the estimation performance by incorporating different number of auxiliary tasks, we denote $\widehat{\beta}_{\widehat{w}^*}(k)$ as the FEASIBLE estimator using the first k auxiliary tasks. For example, $\widehat{\beta}_{\widehat{w}^*}(10)$ is the FEASIBLE estimator only using all useful auxiliary tasks, and $\widehat{\beta}_{\widehat{w}^*}(k)$ with $k > 10$ is the FEASIBLE estimator using both useful and useless auxiliary tasks. The MSE values are then computed and reported in Figure 4. From Figure 4, an interesting U-shape pattern is detected. Specifically, we find that the MSE values decrease as k increases for $1 \le k \le 10$. This is expected since the auxiliary tasks included at this stage are all useful with $w_k^* \neq 0$. However, once $k > 10$, the MSE value starts to increase steadily as k increases. This is also expected since the newly included tasks at this stage are of low quality with $w_k^* = 0$. This suggests that including low-quality tasks might lead to degraded estimation efficiency.

3.6 An Extension: Logistic Regression

In this section, we evaluate the possibility of extending our method from a linear regression model to some other models. To illustrate this idea, we consider the popularly used logistic regression model as an example. We first follow the data generation process in Section 3.1 to generate the feature vector X_i and the coefficient matrix B. For simplicity, we assume a

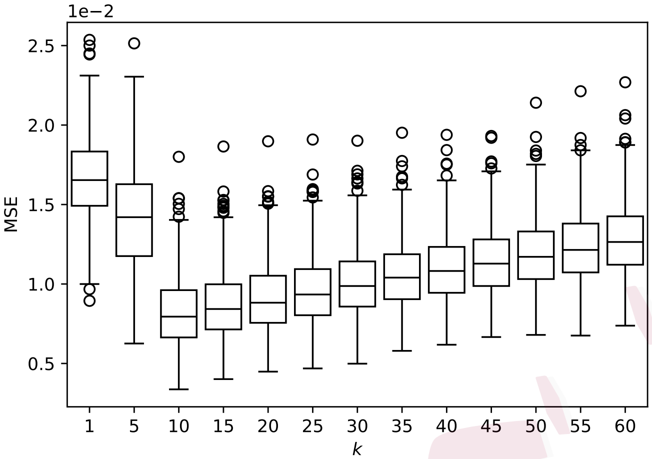
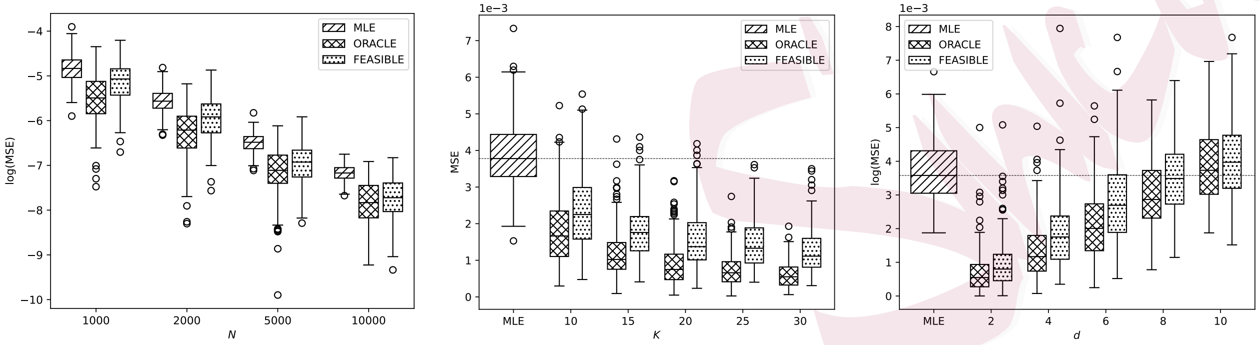


Figure 4: Boxplots of MSE values of FEASIBLE estimators using different numbers of auxiliary tasks.

diagonal covariance matrix Σ_ε for the random noise. That is, $\Sigma_\varepsilon = \text{diag}(\sigma_k^2 : 0 \le k \le K) \in \mathbb{R}^{(K+1)\times(K+1)}$. Then assume the response $Y_i^{(k)}$ with $0 \le k \le K$ is a binary variable, which is generated from a Bernoulli distribution with the probability given as $p_{ik} = P(Y_i^{(k)} = 1|X_i, \beta^{(k)}) = \exp(X_i^\top \beta^{(k)})/\{1 + \exp(X_i^\top \beta^{(k)})\}$. The data generation process is replicated for $M = 500$ times. Then we compare the ORACLE and FEASIBLE estimators with the maximum likelihood estimator (MLE) only using the primary response $Y_i^{(0)}$ and X_i. For the ORACLE and FEASIBLE estimators, the weights w^* and $\widehat{w}^*$ are calculated similarly as those in the linear regression model. The only difference lies in the computation of variance terms. That is $\widehat{\Sigma}_\varepsilon = \text{diag}(\widehat{\sigma}_k^2) \in \mathbb{R}^{(K+1)\times(K+1)}$, where $\widehat{\sigma}_k^2 = N^{-1}\sum_{i=1}^N \widehat{p}_{ik}(1-\widehat{p}_{ik})$. Here $\widehat{p}_{ik}$ is the estimated counterpart of p_{ik} with $\beta^{(k)}$s replaced by the corresponding MLE.

We consider different combinations of (N, p, K, d) to explore the estimation performance of the weighted estimators. Specifically, for varying N and p, we fix $K = 10$ and $d = 5$. For varying K and fixed d, we assume $N = 2000$, $p = \sqrt{N}$, and $d = 5$. For varying d and fixed K, assume $K = 10$, $N = 2000$, and $p = \sqrt{N}$. The detailed simulation results are shown in Figure 5. In general, the results under the logistic regression model are quantitatively similar to those under the linear regression model. We demonstrate that the ORACLE and FEASIBLE estimators are both consistent as the MSE values decrease with the increase of the sample size. The two weighted estimators also enjoy better statistical efficiency than the MLE as B is rank-reducible.



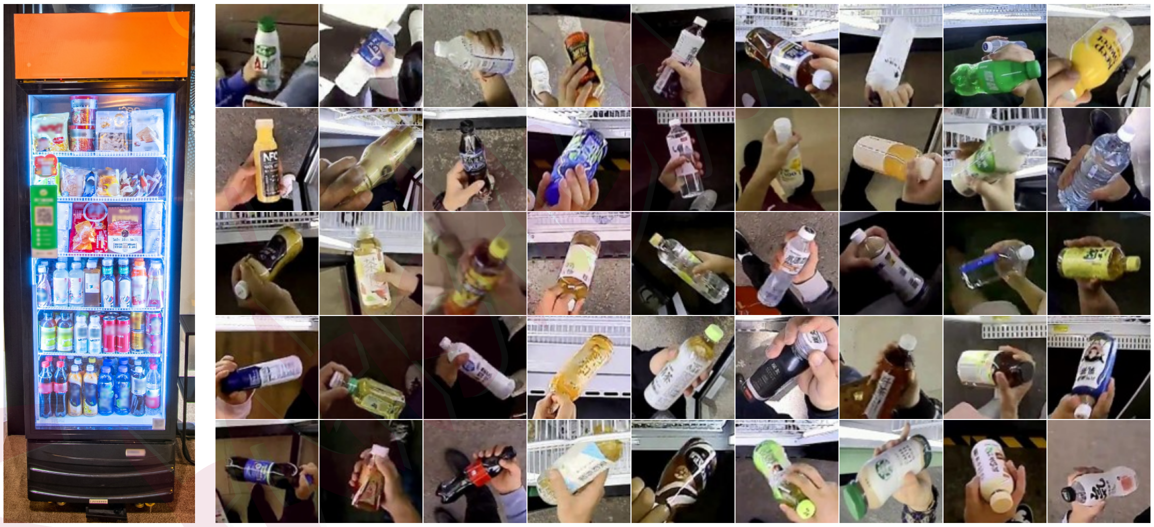
(a) Varying N and p (b) Varying K, Fixed d (c) Varying d, Fixed K

Figure 5: Boxplots of MSE values of the MLE, ORACLE, and FEASIBLE estimators for the logistic regression model. For varying N and p, we fix $K = 10$ and $d = 5$. For varying K, we fix $d = 5$, $N = 2000$, and $p = \sqrt{N}$. For varying d, we fix $K = 10$, $N = 2000$, and $p = \sqrt{N}$

4. REAL DATA ANALYSIS

4.1 Data Description

To demonstrate the practical performance of our method, we conduct an empirical study to identify products in smart vending machines, which are commonly encountered in our daily life. Here we focus on one particular type of smart vending machine as shown in the left panel of Figure 6. With this type of smart vending machine, users can experience a convenient shopping process characterized by "scan to open, choose freely, and automatically settle when closed" (Wang et al., 2022). Each smart vending machine is equipped with a camera that records the shopping process of users. Then based on the video recordings, the smart vending machine needs to accurately identify the products purchased by the user and then establish financial settlement. Thus in this work, our primary goal is to identify the products. For illustration purposes, we choose the product type of "bottled water", which covers a total of forty-five categories (see the right panel of Figure 6). The dataset is provided by a Chinese smart retail company, which consists of $N = 6,018$ transactions for bottled water. For each transaction, a sub-image containing the purchased product is carefully sampled from the camera recordings by an appropriately designed automatic approach. This leads to $N = 6,018$ photos, each of which corresponds to a unique transaction. In the subsequent analysis, we aim to identify the products based on the photo information.



(a) Smart vending machine

(b) 45 types of bottled water

Figure 6: The left panel is one smart vending machine and the right panel shows 45 types of bottled water sold at the vending machine.

4.2 The Design of Auxiliary Tasks

We treat this product classification problem as a multi-class classification task. Specifically, define the primary response as the product category labeled as $Y_i^{(0)} \in \{1, \cdots, 45\}$. Based on some preliminary exploration, directly classifying $Y_i^{(0)}$ does not yield satisfactory results. To improve the classification accuracy, we seek the help of auxiliary learning. Specifically, we consider five auxiliary tasks. The first one is identifying the cap color of the bottle (e.g., white, blue, red). It is a ten-class task and thus $Y_i^{(1)} \in \{1, \cdots, 10\}$. The second one is identifying the label shape of the bottle (e.g., wide, narrow, wrapped). It is a five-class task and thus we have $Y_i^{(2)} \in \{1, \cdots, 5\}$. The third one is identifying the liquid color (e.g., white, black, yellow). It is a nine-class task and thus $Y_i^{(3)} \in \{1, \cdots, 9\}$. The fourth task is to identify whether there are horizontal ridges on the plastic bottle. This is a binary label and $Y_i^{(4)} \in \{1, 2\}$. The final task is to identify if there are vertical ridges, which is also a binary label as $Y_i^{(5)} \in \{1, 2\}$. See Figure 7 for the aforementioned auxiliary tasks of two specific products for visualization. The detailed categories of the five tasks can be found in Figure 8.

4.3 Model Training and Performance Measures

We randomly split the whole dataset into a training set and a testing set, whose index sets are denoted by $\mathcal{I}_{\text{train}}$ and $\mathcal{I}_{\text{test}}$, respectively. To handle the photo information, we adopt the DenseNet architecture (Huang et al., 2017) for both the primary task and five auxiliary tasks. First, we resize the input images to dimensions of 224 pixels for both height and width. Subsequently, we initialize the convolutional layers of the model using pre-trained weights provided by the open-source framework PyTorch (Paszke et al., 2017). After this

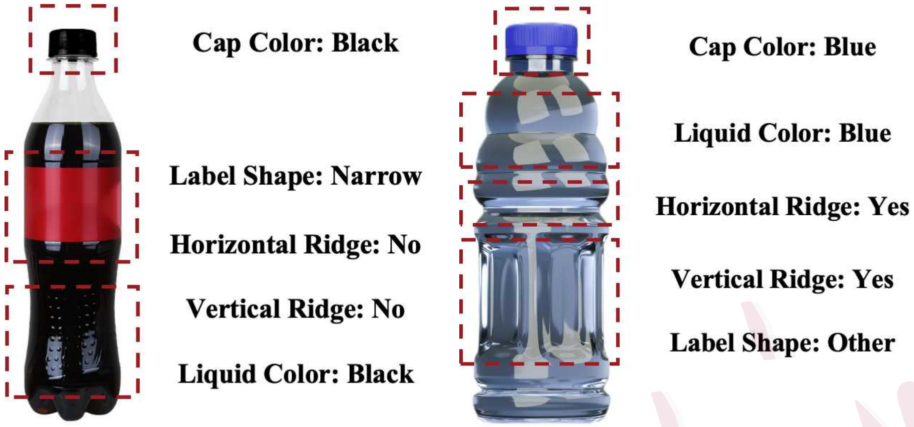
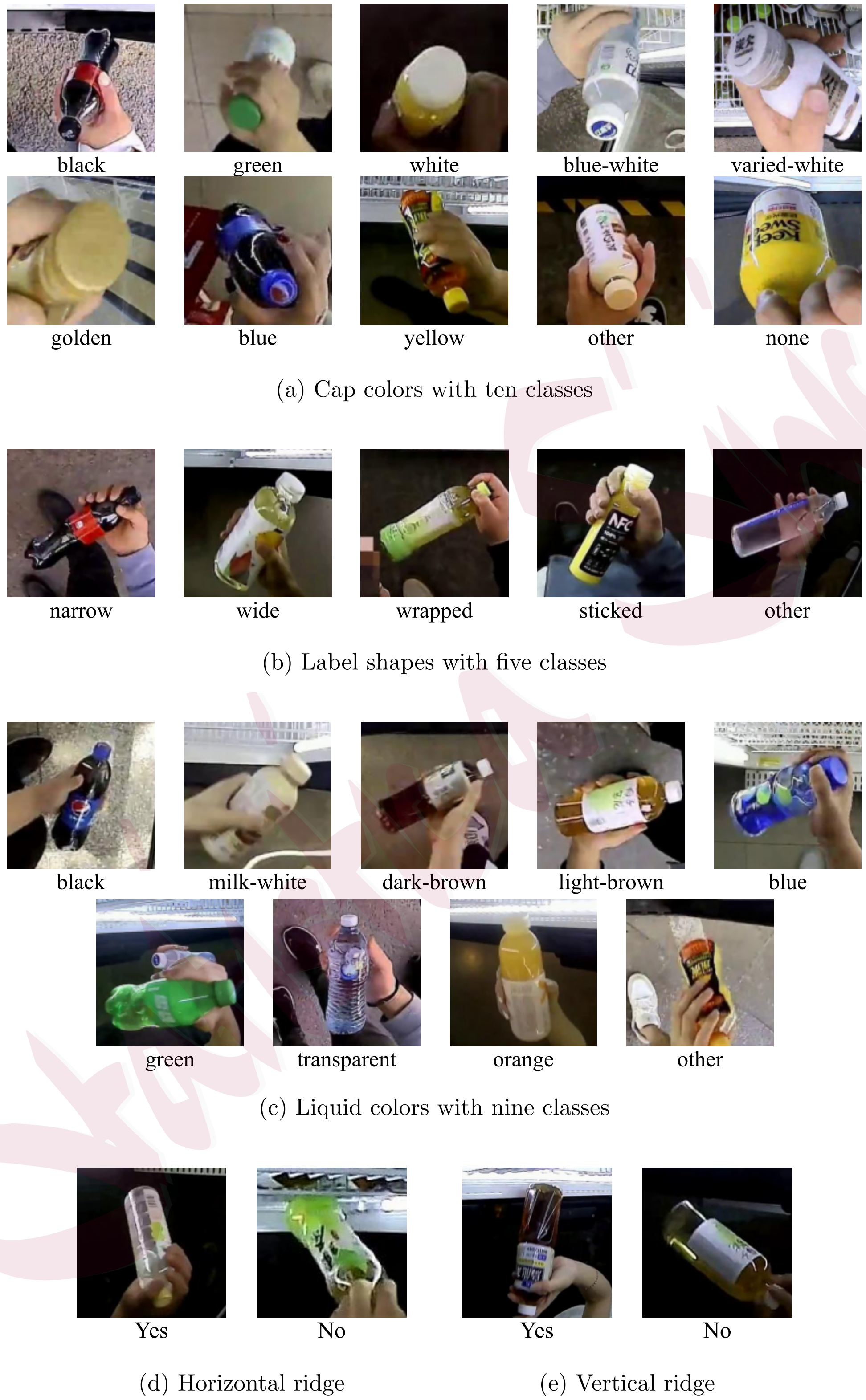


Figure 7: The illustration of two specific products. The product on the left panel is with "black" cap color, "narrow" label shape, and "black" liquid color. It does not have horizontal or vertical ridges. The product on the right panel is with "blue" cap color, "blue" liquid color, and "other" label shape. It has both horizontal and vertical ridges.

pre-processing, the model should generate a 1920-dimensional feature vector after passing through the convolutional layers. Then for the primary task, this feature vector undergoes mapping through a fully connected layer, which results in a total of 45 coefficient vectors. A softmax transformation is then applied to obtain the predicted probabilities. Similarly, for each auxiliary task, the same feature vector is mapped through the corresponding fully connected layer with a softmax transformation. To train the model, we employ the cross-entropy loss and utilize the Adam optimizer with a fixed learning rate of 10^{-4} so that the whole model parameters can be fine-tuned on our dataset. Similar operations are conducted for the auxiliary tasks.

After model training, we adopt the classification error rate (Err) to evaluate the prediction performance for the primary task on the testing set. Specifically, denote the sample size of the testing set as $n^* = |\mathcal{I}_{\text{test}}|$. Then the classification error rate is calculated as $\text{Err} = (n^*)^{-1} \sum_{i \in \mathcal{I}_{\text{test}}} \mathbb{I}(Y_i^{(0)} \neq \widehat{Y}_i^{(0)})$, where $\mathbb{I}(\cdot)$ is an indicator function. In this work, we adopt the Err metric primarily because, in real smart vending machine operations, the aver-



(a) Cap colors with ten classes

(b) Label shapes with five classes

(c) Liquid colors with nine classes

(d) Horizontal ridge

(e) Vertical ridge

Figure 8: The detailed classification categories of the five auxiliary tasks. Each item has a label in each of the five auxiliary classification tasks.

age revenue from each correct transaction settlement is quite limited, but the loss resulting from an incorrect transaction can be substantial. In case of a wrong transaction settlement, some customers may apply for a refund from the merchant, resulting in minimal loss for the merchant. However, other customers might directly file complaints with authoritative regulatory agencies for compensation, typically leading to a penalty of 10 times the original price of the goods involved in the transaction. Note that the profit from a single correct transaction is no more than 10% of the original price. This implies that the loss from one incorrect transaction settlement could require at least $10/0.1 = 100$ correct transactions to compensate. This makes the classification error rate the most important performance metric used in smart vending machine operations.

4.4 Task and Rank Selection

For a practical application, the auxiliary tasks and the rank of the coefficient matrix need to be selected. Intuitively, only those auxiliary tasks truly helpful for reducing the final classification error should be included. Those redundant auxiliary tasks should be excluded. Otherwise, the classification error rate can be inflated. The rank of the coefficient matrix should also be carefully estimated. On one hand, the estimated rank d should not be too small. Recall that the optimal weight is contained in $\mathbb{W}$. Once d is under-estimated, the resulting weight might not be contained in $\mathbb{W}$. Hence the resulting weighted estimator would be seriously biased. On the other hand, the estimated rank cannot be too large either. Once d is over-estimated, the resulting weight might not be optimal, and hence $\mathcal{P}(w^*)$ could be sub-optimal. Consequently, the resulting weighted estimator might suffer unnecessarily large variability. Therefore, both the auxiliary tasks and the rank of the coefficient matrix should be carefully selected. To this end, we develop here a two-step cross-validation method for task and rank selection.

We start with the task selection. We temporarily assume the coefficient matrix B is of full rank. This is a case with d potentially over-estimated instead of under-estimated. Then the resulting weighted estimator should be at least unbiased as we mentioned previously. Next, we start with the full task set $\mathcal{S}_0 = \{1, 2, 3, 4, 5\}$, which collects the indices of all auxiliary tasks. This represents the case with all auxiliary tasks being used for model training. We next randomly split the whole dataset into $\mathcal{I}_{\text{train}}$ and $\mathcal{I}_{\text{test}}$. Recall that $\mathcal{I}_{\text{train}}$ stands for the indices of the training dataset, which accounts for about 80% of the whole dataset. Then the initial model can be computed based on $\mathcal{S}_0$ and $\mathcal{I}_{\text{train}}$. The corresponding classification error rate is then evaluated on the $\mathcal{I}_{\text{test}}$, which accounts for the rest 20% of the whole dataset. We replicate the experiment for a total of 50 times by randomly partitioning the training and testing sets. This leads to a total of 50 Err values, which are then averaged and recorded as AvgErr_0.

For each $k \in \mathcal{S}_0$, we consider a reduced task set as $\mathcal{S}_{(0,k)} = \mathcal{S}_0 \backslash \{k\}$. We can then evaluate the classification error rate of $\mathcal{S}_{(0,k)}$ by a similar method as for $\mathcal{S}_0$. The classification error rates in 50 replications are then averaged as $\text{AvgErr}_{(0,k)}$ for each $\mathcal{S}_{(0,k)}$. Next define $k_1^* = \arg\min_{k \in \mathcal{S}_0} \text{AvgErr}_{(0,k)}$. This suggests that the auxiliary task indexed by k_1^* should be excluded from $\mathcal{S}_0$. This leads to an optimal choice of the auxiliary task set with size $(K-1)$ as $\mathcal{S}_1 = \mathcal{S}_0 \backslash \{k_1^*\}$. The corresponding AvgErr value is then denoted by $\text{AvgErr}_1 = \text{AvgErr}_{(0,k_1^*)}$. We then repeat this process for every $\mathcal{S}_k$ so that $\mathcal{S}_{k+1}$ can be constructed for $0 \leq k < 5$, each of which is associated with AvgErr_{k+1}. This leads to a sequence of nested task sets $\mathcal{S}_k$ with $0 \leq k \leq 5$. Then the best auxiliary task set is given by $\mathcal{S}_{\text{opt}} = \mathcal{S}_{k_{\text{opt}}}$, where

$k_{\text{opt}} = \arg\min_k \text{AvgErr}_k$. By applying this method to our smart-vending-machine dataset, we obtain the optimal auxiliary task set as $\mathcal{S}_{\text{opt}} = \mathcal{S}_1 = \{1, 2, 3, 4\}$ with $\text{AvgErr}_1 = 0.0468$.

With the chosen $\mathcal{S}_{\text{opt}} = \{1, 2, 3, 4\}$, we next focus on the rank selection. Specifically, we aim to investigate the impact of different values of d on the classification error rate. Note that, with the auxiliary task set $\mathcal{S}_{\text{opt}} = \{1, 2, 3, 4\}$, the coefficient matrix B is a $p \times K_0$ matrix, where $K_0 = 45 + 10 + 5 + 9 + 2 = 71$ is the total number of categories in the primary task and four auxiliary tasks. Therefore, we vary $d = \text{rank}(B)$ in the range $[1, 71]$. For each given d, we compute the weighted estimator for the primary task on the training dataset $\mathcal{I}_{\text{train}}$. Then its classification error rate is evaluated on $\mathcal{I}_{\text{test}}$. This process is also randomly replicated for a total of 50 times for each d. The resulting 50 Err values are then averaged as $\text{AvgErr}^{(d)}$ for each d. We then report $\text{AvgErr}^{(d)}$ in the left panel of Figure 9.

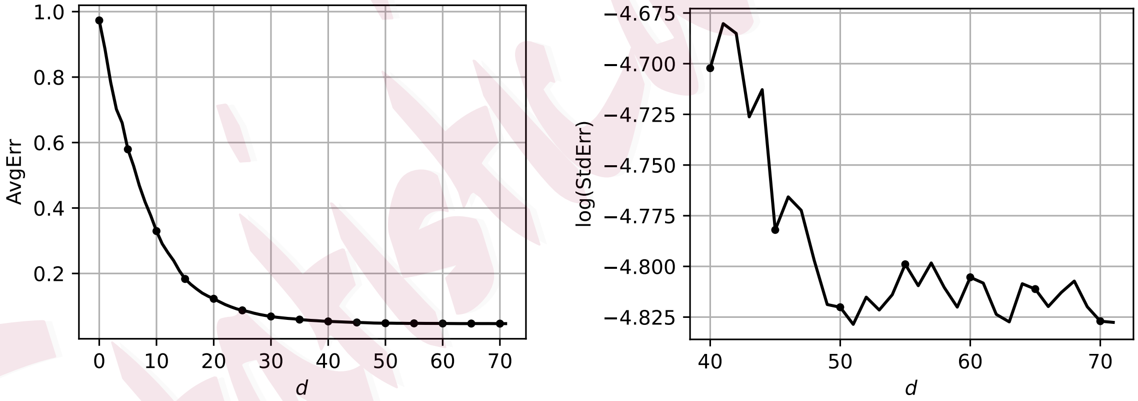


Figure 9: The detailed results of AvgErr and StdErr (in logarithm) under different ranks.

From Figure 9, we find a rather flat pattern for $\text{AvgErr}^{(d)}$ as $d \in [40, 71]$. This suggests that the classification error rate is fairly insensitive for a broad range of $d \in [40, 71]$. Within this specification range of d, we also study the stability of the classification error rate by calculating the standard deviation of Errs, which is denoted as $\text{StdErr}^{(d)}$. We report this result in the right panel of Figure 9. We find that the minimum of $\text{StdErr}^{(d)}$ is achieved at

$d = 51$. Therefore, we should consider $d = 51$ as the final specification about $\text{rank}(B)$, as it yields almost the lowest classification error rate and the greatest stability.

4.5 Model Comparison

We compare the classification performance of our weighted estimator with that of other methods. With $\mathcal{S}_{\text{opt}} = \{1, 2, 3, 4\}$ and $d = 51$, the final classification error rate of our method is given by AvgErr $= 0.0481$. For comparison purpose, we consider a total number of five competitors. The first one is a benchmark model only using the primary task, which is denoted by STL (single-task learning). The second one is a model using all five auxiliary tasks, which can be considered as a special case of multi-task learning and thus denoted by MTL. Three auxiliary learning algorithms are also included for evaluation. They are, respectively, the AL-MTL method (Liebel and Körner, 2018), the AuxiLearn method (Navon et al., 2020), and the JTDS method (Chen et al., 2022). These three methods focus on obtaining the estimator by optimizing the weighted loss functions. For each method, the experiments are replicated for 50 times. The classification error rates of different methods are summarized in Table 1. The corresponding standard deviations (StdErr) for different methods are also reported. Additionally, we calculate the Relative Improvement (RI) in Errs between each competitor and our weighted estimator. Take the STL method as an example. The relative improvement is computed as $(1-0.0481/0.0535)\times100\% = 10.09\%$. Similarly, the RI value for the MTL method verse our method is computed as $(1-0.0481/0.0534)\times100\% = 9.93\%$.

From Table 1, we can draw the following conclusions. First, the classification error rates of three deep-learning based methods (i.e., AL-MTL, AuxiLearn, and JTDS) are much higher

Table 1: The comparison results with different competitive methods.

Methods	AvgErr	StdErr	RI
Weighted Estimator (Ours)	0.0481	0.0079	–
Single-Task Learning (STL)	0.0535	0.0211	10.09%
Multi-Task Learning (MTL)	0.0534	0.0151	9.93%
AL-MTL (Liebel and Körner, 2018)	0.0808	0.0062	40.47%
AuxiLearn (Navon et al., 2020)	0.0842	0.0251	42.87%
JTDS (Chen et al., 2022)	0.0766	0.0048	37.20%

than those of the other methods. This finding suggests that, deep-learning methods are not applicable in our smart-vending-machine application. Second, the classification error rate of our proposed weighted estimator outperforms that of STL and MTL. This result implies that, using carefully selected auxiliary tasks can benefit the prediction performance. Last, we find the standard deviation of our proposed method is much smaller than that of STL and MTL. This is consistent with our theoretical results in Corollary 1, since our method enjoys the optimal $\mathcal{P}(w^*)$ and thus has lower variability.

5. CONCLUDING REMARKS

Auxiliary learning is a popular method in machine learning research. The key idea of auxiliary learning is to introduce a set of auxiliary tasks to enhance the performance of the primary task. In this work, we adopt the idea of auxiliary learning to address the parameter estimation problem in high-dimensional settings. We start with the linear regression model. To improve the statistical efficiency of the parameter of primary research, we develop a weighted estimator, which is a linear combination of the OLS estimators of both the primary task and auxiliary tasks. The optimal weight is analytically derived and the statistical properties of the corresponding weighted estimator are studied. The weighted estimator is then extended to generalized linear regression models. Extensive numerical studies are presented to demonstrate the finite sample performance of the proposed method. Last, we apply the weighted estimator to address the product classification problem of smart vending machines. Results show that by using auxiliary learning, the products can be predicted more accurately than several state-of-the-art methods.

To conclude this article, we consider here some interesting topics for future studies. First, we demonstrate that the proposed ORACLE and FEASIBLE estimators outperform the traditional OLS estimator by achieving lower mean squared errors. Our numerical studies suggest the usefulness of the weighted estimator in generalized linear models. However, the theoretical analysis under generalized linear models or more general loss functions needs further investigation. Second, although we demonstrate that auxiliary learning can enhance estimation accuracy and efficiency, designing high-quality auxiliary tasks remains a significant challenge. This is an interesting topic for future exploration.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.72371241, 72171229), Beijing Social Science Fund (24GLC033), and the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD910001).

Supplementary Materials

Appendix A provides the detailed verification for the form of $\mathbb{W}$ and w^*. Appendix B provides the proof of Theorem 1. Appendix C provides some useful lemmas for Theorem 1.

References

Chen, H., X. Wang, C. Guan, Y. Liu, and W. Zhu (2022). Auxiliary learning with joint task and data scheduling. In *International Conference on Machine Learning*, pp. 3634–3647. PMLR.

Chen, H., X. Wang, Y. Liu, Y. Zhou, C. Guan, and W. Zhu (2022). Module-aware optimization for auxiliary learning. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, pp. 145–152.

Chen, Z., V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich (2018). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR.

Fan, J., Y. Fan, and J. Lv (2008). High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics 147*(1), 186–197.

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

Fan, J., R. Li, C.-H. Zhang, and H. Zou (2020). *Statistical foundations of data science.* CRC press.

Fan, J., Z. Lou, and M. Yu (2024). Are latent factor regression and sparse regression adequate? *Journal of the American Statistical Association 119*(546), 1076–1088.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(5), 849–911.

Gao, Y., W. Liu, H. Wang, X. Wang, Y. Yan, and R. Zhang (2022). A review of distributed statistical inference. *Statistical Theory and Related Fields 6*(2), 89–99.

Howard, J. and S. Ruder (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Huang, G., Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Jiang, Y., Y. He, and H. Zhang (2016). Variable selection with prior information for generalized linear models via the prior lasso method. *Journal of the American Statistical Association 111*(513), 355–376.

Jiang, Z., H. Guo, and J. Wang (2023). Feature screening for multiple responses. *Journal of Multivariate Analysis 198*, 105223.

Kanezaki, A., Y. Matsushita, and Y. Nishida (2018). Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5010–5019.

Kendall, A., Y. Gal, and R. Cipolla (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491.

Kokkinos, I. (2017). Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6129–6138.

Kung, P.-N., S.-S. Yin, Y.-C. Chen, T.-H. Yang, and Y.-N. Chen (2021, November). Efficient multi-task auxiliary learning: Selecting auxiliary data by feature similarity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, pp. 416–428. Association for Computational Linguistics.

Li, B. and A. Dong (2021). Multi-task learning with attention : Constructing auxiliary tasks for learning to learn. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 145–152.

Li, R., W. Zhong, and L. Zhu (2012). Feature screening via distance correlation learning. *Journal of the American Statistical Association 107*(499), 1129.

Li, X., Y. Gao, H. Chang, D. Huang, Y. Ma, R. Pan, H. Qi, F. Wang, S. Wu, K. Xu, et al. (2024). A selective review on statistical methods for massive data computation: distributed computing, subsampling, and minibatch techniques. *Statistical Theory and Related Fields*, 1–23.

Liebel, L. and M. Körner (2018). Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334*.

Misra, I., A. Shrivastava, A. Gupta, and M. Hebert (2016). Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003.

Murray, J. D. (2012). *Asymptotic analysis*, Volume 48. Springer Science & Business Media.

Navon, A., I. Achituve, H. Maron, G. Chechik, and E. Fetaya (2020). Auxiliary learning by implicit differentiation. *arXiv preprint arXiv:2007.02693*.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer (2017). Automatic differentiation in pytorch.

Ruder, S., J. Bingel, I. Augenstein, and A. Søgaard (2019). Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 33, pp. 4822–4829.

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

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

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

Vandenhende, S., S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool (2021). Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence 44*(7), 3614–3633.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, Volume 47. Cambridge university press.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, Volume 48. Cambridge university press.

Wang, F., J. Liu, and H. Wang (2021). Sequential text-term selection in vector space models. *Journal of Business & Economic Statistics 39*(1), 82–97.

Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association 104*(488), 1512–1524.

Wang, H. (2012). Factor profiled sure independence screening. *Biometrika 99*(1), 15–28.

Wang, X., T. Li, J. Chen, S. Zong, and Z. Kai (2022). Design and implementation of intelligent unmanned vending container system. *Modern Electronics Technique 6*, 163–168,.

Yuan, M. and Y. Lin (2005). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology 68*(1), 49–67.

Zhu, X., D. Huang, R. Pan, and H. Wang (2020). Multivariate spatial autoregressive model for large scale social networks. *Journal of Econometrics 215*(2), 591–606.

Zhu, X., R. Pan, S. Wu, and H. Wang (2022). Feature screening for massive data analysis by subsampling. *Journal of Business & Economic Statistics 40*(4), 1892–1903.

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

Guanghua School of Management, Peking University, Beijing, China.

E-mail: hanchao.yan@stu.pku.edu.cn

Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China

E-mail: feifei.wang@ruc.edu.cn

Chinese Academy of International Trade and Economic Cooperation, Ministry of Commerce People's Republic of China, Beijing, China.

E-mail: xiachuanxin@caitec.org.cn

Guanghua School of Management, Peking University, Beijing, China.

E-mail: hansheng@pku.edu.cn