Published in final edited form as:
J Am Stat Assoc. 2023 ; 118(544): 2684–2697. doi:10.1080/01621459.2022.2071278.

Transfer Learning under High-dimensional Generalized Linear Models

Ye Tian,
Department of Statistics, Columbia University

Yang Feng
Department of Biostatistics, School of Global Public Health, New York University

Abstract

In this work, we study the transfer learning problem under highdimensional generalized linear models (GLMs), which aim to improve the fit on *target* data by borrowing information from useful *source* data. Given which sources to transfer, we propose a transfer learning algorithm on GLM, and derive its ℓ_1 / ℓ_2-estimation error bounds as well as a bound for a prediction error measure. The theoretical analysis shows that when the target and source are sufficiently close to each other, these bounds could be improved over those of the classical penalized estimator using only target data under mild conditions. When we don't know which sources to transfer, an *algorithm-free* transferable source detection approach is introduced to detect informative sources. The detection consistency is proved under the high-dimensional GLM transfer learning setting. We also propose an algorithm to construct confidence intervals of each coefficient component, and the corresponding theories are provided. Extensive simulations and a real-data experiment verify the effectiveness of our algorithms. We implement the proposed GLM transfer learning algorithms in a new R package `glmtrans`, which is available on CRAN.

Keywords

Generalized linear models; transfer learning; high-dimensional inference; Lasso; sparsity; negative transfer

1. Introduction

Nowadays, a great deal of machine learning algorithms has been successfully applied in our daily life. Many of these algorithms require sufficient training data to perform well, which sometimes can be limited. For example, from an online merchant 's view, it could be difficult to collect enough personal purchase data for predicting the customers' purchase behavior and recommending corresponding items. However, in many cases, some related datasets may be available in addition to the limited data for the original task. In the merchant-customer example, we may also have the customers' clicking data in hand, which is not exactly the same as but shares similarities with the purchase data. How to use these

yang.feng@nyu.edu .

additional data to help with the original target task motivates a well-known concept in computer science: *transfer learning* (Torrey and Shavlik, 2010; Weiss et al., 2016). As its name indicates, in a transfer learning problem, we aim to transfer some useful information from similar tasks (*sources*) to the original task (*target*), in order to boost the performance on the target. To date, transfer learning has been widely applied in a number of machine learning applications, including the customer review classification (Pan and Yang, 2009), medical diagnosis (Hajiramezanali and Zamani, 2018), and ride dispatching in ride-sharing platforms (Wang et al., 2018), etc. Compared with the rapidly growing applications, there has been little discussion about the theoretical guarantee of transfer learning. Besides, although transfer learning has been prevailing in computer science community for decades, far less attention has been paid to it among statisticians. More specifically, transfer learning can be promising in the *high-dimensional* data analysis, where the sample size is much less than the dimension with some sparsity structure in the data (Tibshirani, 1996). The impact of transfer learning in high-dimensional generalized linear models (GLMs) with sparsity structure is not quite clear up to now. In this paper, we are trying to fill the gap by developing transfer learning tools in high-dimensional GLM inference problem, and providing corresponding theoretical guarantees.

Prior to our paper, there are a few pioneering works exploring transfer learning under the high-dimensional setting. Bastani (2021) studied the single-source case when the target data comes from a high-dimensional GLM with limited sample size while the source data size is sufficiently large than the dimension. A two-step transfer learning algorithm was developed, and the estimation error bound was derived when the contrast between target and source coefficients is ℓ_0-sparse. Li et al. (2021) further explored the multi-source high-dimensional linear regression problem where both target and source samples are high-dimensional. The ℓ_2-estimation error bound under ℓ_q-regularization ($q \in [0, 1]$) was derived and proved to be minimax optimal under some conditions. In Li et al. (2020), the analysis was extended to the Gaussian graphical models with false discovery rate control. Other related research on transfer learning with theoretical guarantee includes the non-parametric classification model (Cai and Wei, 2021; Reeve et al., 2021) and the analysis under general functional classes via transfer exponents (Hanneke and Kpotufe, 2020a,b), etc. In addition, during the past few years, there have been some related works studying parameter sharing under the regression setting. For instance, Chen et al. (2015) and Zheng et al. (2019) developed the so-called “data enriched model” for linear and logistic regression under a single-source setting, where the properties of the oracle tuned estimator with a quadratic penalty were studied. Gross and Tibshirani (2016) and Ollier and Viallon (2017) explored the so-called “data shared Lasso” under the multi-task learning setting, where ℓ_1 penalties of all contrasts are considered.

In this work, we contribute to transfer learning under a high-dimensional context from three perspectives. First, we extend the results of Bastani (2021) and Li et al. (2021), by proposing multi-source transfer learning algorithms on generalized linear models (GLMs) and we assume both target and source data to be high-dimensional. We assume the contrast between target and source coefficients to be ℓ_1-sparse, which differs from the ℓ_0-sparsity considered in Bastani (2021). The theoretical analysis shows that when the target and source are sufficiently close to each other, the estimation error bound of target coefficients could be improved over that of the classical penalized estimator using only target data under mild conditions. Moreover, the error rate is shown to be minimax optimal under certain conditions. To the best of our knowledge, this is the first study of the multi-source transfer learning framework under the high-dimensional GLM setting. Second, as we mentioned, transferring sources that are close to the target can bring benefits. However, some sources might be far away from the target, and transferring them can be harmful. This phenomenon is often called *negative transfer* in literature (Torrey and Shavlik, 2010). We will show the impact of negative transfer in simulation studies in Section 4.1. To avoid this issue, we develop an *algorithm-free* transferable source detection algorithm, which can help identify informative sources. And with certain conditions satisfied, the algorithm is shown to be able to distinguish useful sources from useless ones. Third, all aforementioned works of transfer learning on high-dimensional regression only focus on the point estimate of the coefficient, which is not sufficient for statistical inference. How transfer learning can benefit the confidence interval construction remains unclear. We propose an algorithm on the basis of our two-step transfer learning procedure and nodewise regression (Van de Geer et al., 2014), to construct the confidence interval for each coefficient component. The corresponding asymptotic theories are established.

The rest of this paper is organized as follows. Section 2 first introduces GLM basics and transfer learning settings under high-dimensional GLM, then presents a general algorithm (where we know which sources are useful) and the transferable source detection algorithm (where useful sources are automatically detected). At the end of Section 2, we develop an algorithm to construct confidence intervals. Section 3 provides the theoretical analysis on the algorithms, including ℓ_1 and ℓ_2-estimation error bounds of the general algorithm, detection consistency property of the transferable source detection algorithm, and asymptotic theories for the confidence interval construction. We conduct extensive simulations and a real-data study in Section 4, and the results demonstrate the effectiveness of our GLM transfer learning algorithms. In Section 5, we review our contributions and shed light on some interesting future research directions. Additional simulation results and theoretical analysis, as well as all the proofs, are relegated to supplementary materials.

2. Methodology

We first introduce some notations to be used throughout the paper. We use bold capitalized letters (e.g. $\boldsymbol{X}$, $\boldsymbol{A}$) to denote matrices, and use bold little letters (e.g. $\boldsymbol{x}$, $\boldsymbol{y}$) to denote vectors. For a p-dimensional vector $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ we denote its ℓ_q-norm as $\| \boldsymbol{x} \| = (\sum_{i=1}^{p} | x_i |^q)^{1/q} (q \in (0,2])$, and ℓ_0-"norm" $\| \boldsymbol{x} \|_0 = \#\{j : x_j \neq 0\}$. For a matrix $\mathbf{A}_{p \times q} = [a_{ij}]_{p \times q}$, its 1-norm, 2-norm, ∞-norm and max-norm are defined as $\| \boldsymbol{A} \|_1 = \sup_j \sum_{i=1}^{p} | a_{ij} |$, $\| \boldsymbol{A} \|_2 = \max_{\boldsymbol{x}: \|\boldsymbol{x}\|_2 = 1} \| \boldsymbol{A}\boldsymbol{x} \|_2$, $\| \boldsymbol{A} \|_\infty = \sup_i \sum_{j=1}^{q} | a_{ij} |$ and $\| \boldsymbol{A} \|_{\max} = \sup_{i,j} | a_{ij} |$, respectively. For two non-zero real sequences $\{ a_n \}_{n=1}^{\infty}$ and $\{ b_n \}_{n=1}^{\infty}$, we use $a_n \ll b_n$, $b_n \gg a_n$ or $a_n = \mathcal{O}(b_n)$ to represent $| a_n / b_n | \to 0$ as $n \to \infty$. And $a_n \lesssim b_n$ or $a_n = \mathcal{O}(b_n)$ means $\sup_n | a_n / b_n | < \infty$. Expression $a_n \asymp b_n$ means that a_n / b_n converges to some positive constant. For two random variable sequences $\{ x_n \}_{n=1}^{\infty}$ and $\{ y_n \}_{n=1}^{\infty}$, notation $x_n \lesssim_p y_n$ or $x_n = \mathcal{O}_p(y_n)$ means that for any $\epsilon > 0$, there exists a positive constant M such that $\sup_n \mathbb{P}(|x_n/y_n| > M) \le \epsilon$. And for two real numbers a and b, we use $a \vee b$ and $a \wedge b$ to represent $\max(a, b)$ and $\min(a, b)$, respectively. Without specific notes, the expectation $\mathbb{E}$, variance Var, and covariance Cov are calculated based on all randomness.

2.1 Generalized linear models (GLMs)

Given the predictors $\boldsymbol{x} \in \mathbb{R}^p$, if the response y follows the generalized linear models (GLMs), then its conditional distribution takes the form

$$y \mid \boldsymbol{x} \sim \mathbb{P}(y \mid \boldsymbol{x}) = \rho(y) \exp\{y\boldsymbol{x}^T\boldsymbol{w} - \psi(\boldsymbol{x}^T\boldsymbol{w})\},$$

where $\boldsymbol{w} \in \mathbb{R}^p$ is the coefficient, ρ and ψ are some known univariate functions. $\psi'(\boldsymbol{x}^T\boldsymbol{w}) = \mathbb{E}(y \mid \boldsymbol{x})$ is called the *inverse link function* (McCullagh and Nelder, 1989). Another important property is that $\mathrm{Var}(y \mid \boldsymbol{x}) = \psi''(\boldsymbol{x}^T\boldsymbol{w})$, which follows from the fact that the distribution belongs to the exponential family. It is ψ that characterizes different GLMs. For example, in linear model with Gaussian noise, we have a continuous response y and $\psi(u) = \frac{1}{2}u^2$; in the logistic regression model, y is binary and $\psi(u) = \log(1 + e^u)$; and in Poisson regression model, y is a nonnegative integer and $\psi(u) = e^u$. For most GLMs, ψ is strictly convex and infinitely differentiable.

2.2 Target data, source data, and transferring level

In this paper, we consider the following multi-source transfer learning problem. Suppose we have the *target* data set $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$ and K *source* data sets with the k-th source denoted as $(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})$, where $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n_k \times p}$, $\boldsymbol{y}^{(k)} \in \mathbb{R}^{n_k}$ for $k = 0, \ldots, K$. The i-th row of $\boldsymbol{X}^{(k)}$ and the i-th element of $\boldsymbol{y}^{(k)}$ are denoted as $\boldsymbol{x}_i^{(k)}$ and $y_i^{(k)}$, respectively. The goal is to transfer useful information from source data to obtain a better model for the target data. We assume the responses in the target and source data all follow the generalized linear model:

$$y^{(k)} \mid \boldsymbol{x} \sim \mathbb{P}(y \mid \boldsymbol{x}) = \rho(y) \exp\{y\boldsymbol{x}^T\boldsymbol{w}^{(k)} - \psi(\boldsymbol{x}^T\boldsymbol{w}^{(k)})\}, \tag{1}$$

for $k = 0, \ldots, K$, with possibly different coefficient $\boldsymbol{w}^{(k)} \in \mathbb{R}^p$, the predictor $\boldsymbol{x} \in \mathbb{R}^p$, and some known univariate functions ρ and ψ. Denote the target parameter as $\boldsymbol{\beta} = \boldsymbol{w}^{(0)}$. Suppose the target model is ℓ_0-sparse, which satisfies $\|\boldsymbol{\beta}\|_0 = s \ll p$. This means that only s of the p variables contribute to the target response. Intuitively, if $\boldsymbol{w}^{(k)}$ is close to $\boldsymbol{\beta}$, the k-th source could be useful for transfer learning.

Define the k-th contrast $\boldsymbol{\delta}^{(k)} = \boldsymbol{\beta} - \boldsymbol{w}^{(k)}$ and we say $\|\boldsymbol{\delta}^{(k)}\|_1$ is the *transferring level* of source k. And we define the *level-h transferring set* $\mathcal{A}_h = \{k : \|\boldsymbol{\delta}^{(k)}\|_1 \le h\}$ as the set of sources which has transferring level lower than h. Note that in general, h can be any positive values and different h values define different $\mathcal{A}_h$ set. However, in our regime of interest, h shall be reasonably small to guarantee that transferring sources in $\mathcal{A}_h$ beneficial. Denote $n_{\mathcal{A}_h} = \sum_{k \in \mathcal{A}_h} n_k$, $\alpha_k = \frac{n_k}{n_{\mathcal{A}_h} + n_0}$ for $k \in \{0\} \cup \mathcal{A}_h$ and $K_{\mathcal{A}_h} = |\mathcal{A}_h|$.

Note that in (1), we assume GLMs of the target and all sources share the same inverse link function ψ. After a careful examination of our proofs for theoretical properties in Section 3, we find that these theoretical results still hold even when the target and each source have their own function ψ, as long as these GLMs satisfy Assumptions 1 and 3 (to be presented in Section 3.1). It means that transferring information across different GLM families is possible. For simplicity, in the following discussion, we assume all these GLMs belong to the same family and hence have the same function ψ

2.3 Two-step GLM transfer learning

We first introduce a general transfer learning algorithm on GLMs, which can be applied to transfer all sources in a given index set $\mathcal{A}$. The algorithm is motivated by the ideas in Bastani (2021) and Li et al. (2021), which we call a *two-step transfer learning algorithm*. The main strategy is to first transfer the information from those sources by pooling all the data to obtain a rough estimator, then correct the bias in the second step using the target data. More specifically, we fit a GLM with ℓ_1-penalty by pooled samples first, then fit the contrast in the second step using only the target by another ℓ_1-regularization. The detailed algorithm ($\mathcal{A}$-Trans-GLM) is presented in Algorithm 1. The transferring step could be understood as to solve the following equation w.r.t. $\boldsymbol{w} \in \mathbb{R}^p$:

$$\sum_{k \in \{0\} \cup \mathcal{A}} \left[(\boldsymbol{X}^{(k)})^T \boldsymbol{y}^{(k)} - \sum_{i=1}^{n_k} \psi'((\boldsymbol{w})^T \boldsymbol{x}_i^{(k)}) \boldsymbol{x}_i^{(k)} \right] = \boldsymbol{0}_p,$$

which converges to the solution of its population version under certain conditions

$$\sum_{k \in \{0\} \cup \mathcal{A}} \alpha_k \mathbb{E}\left\{ [\psi'((\boldsymbol{w}^{\mathcal{A}})^T \boldsymbol{x}^{(k)}) - \psi'((\boldsymbol{w}^{(k)})^T \boldsymbol{x}^{(k)})] \boldsymbol{x}^{(k)} \right\} = \boldsymbol{0}_p, \tag{2}$$

where $\alpha_k = \frac{n_k}{n_{\mathcal{A}} + n_0}$. Notice that in the linear case, $\boldsymbol{w}^{\mathcal{A}}$ can be explicitly expressed as a linear transformation of the true parameter $\boldsymbol{w}^{(k)}$, i.e. $\boldsymbol{w}^{\mathcal{A}} = \boldsymbol{\Sigma}^{-1} \sum_{k \in \{0\} \cup \mathcal{A}} \alpha_k \boldsymbol{\Sigma}^{(k)} \boldsymbol{w}^{(k)}$, where $\boldsymbol{\Sigma}^{(k)} = \mathbb{E}[\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T]$ and $\boldsymbol{\Sigma} = \sum_{k \in \{0\} \cup \mathcal{A}} \alpha_k \mathbb{E}[\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T]$ (Li et al., 2021).

To help readers better understand the algorithm, we draw a schematic in Section S.1.1 of supplements. We refer interested readers who want to get more intuitions to that.

Algorithm 1: $\mathscr{A}$ -Trans-GLM

```
Input: target data $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$, source data $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}k = 1K$, penalty parameters $\lambda_w$ and $\lambda_\delta$, transferring set $\mathscr{A}$
Output: the estimated coefficient vector $\boldsymbol{\beta}$
1 Transferring step: Compute
    $\boldsymbol{w}^{\mathscr{A}} \leftarrow \arg\min_{\boldsymbol{w}}\left\{\frac{1}{n_{\mathscr{A}} + n_0}\sum_{k \in \{0\} \cup \mathscr{A}}\left[-(\boldsymbol{y}^{(k)})^T \boldsymbol{X}^{(k)}\boldsymbol{w} + \sum_{i=1}^{n_k} \psi(\boldsymbol{w}^T \boldsymbol{x}_i^{(k)})\right] + \lambda_{\boldsymbol{w}} \| \boldsymbol{w} \|_1\right\}$
2 Debiasing step: Compute
    $\boldsymbol{\delta}^{\mathscr{A}} \leftarrow \arg\min_{\boldsymbol{\delta}}\left\{-\frac{1}{n_0}(\boldsymbol{y}^{(0)})^T \boldsymbol{X}^{(0)}(\boldsymbol{w}^{\mathscr{A}} + \boldsymbol{\delta}) + \frac{1}{n_0}\sum_{i=1}^{n_0} \psi((\boldsymbol{w}^{\mathscr{A}} + \boldsymbol{\delta})^T \boldsymbol{x}_i^{(0)}) + \lambda_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_1\right\}$
3 Let $\boldsymbol{\beta} \leftarrow \boldsymbol{w}^{\mathscr{A}} + \boldsymbol{\delta}^{\mathscr{A}}$
4 Output $\boldsymbol{\beta}$
```

2.4 Transferable source detection

As we described, Algorithm 1 can be applied only if we are certain about which sources to transfer, which in practice may not be known as a priori. Transferring certain sources may not improve the performance of the fitted model based on only target, and can even lead to worse performance. In transfer learning, we say *negative transfer* happens when the source data leads to an inferior performance on the target task (Pan and Yang, 2009; Torrey and Shavlik, 2010; Weiss et al., 2016). How to avoid negative transfer has become an increasingly popular research topic.

Here we propose a simple, *algorithm-free,* and *data-driven* method to determine an informative transferring set $\mathscr{A}$. We call this approach a transferable source *detection* algorithm and refer to it as Trans-GLM.

We sketch this detection algorithm as follows. First, divide the target data into three folds, that is, $\{(\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})\}_{r=1}^{3}$. Note that we choose three folds only for convenience. We also explored other fold number choices in the simulation. See Section S.1.3.3 in the supplementary materials. Second, run the transferring step on each source data and every two folds of target data. Then, for a given loss function, we calculate its value on the left-out fold of target data and compute the average cross-validation loss $\widehat{L}_0^{(k)}$ for each source. As a benchmark, we also fit Lasso on every choice of two folds of target data and calculate the loss on the remaining fold. The average cross-validation loss $\widehat{L}_0^{(0)}$ is viewed as the loss of target. Finally, the difference between $\widehat{L}_0^{(k)}$ and $\widehat{L}_0^{(0)}$ is calculated and compared with some threshold, and sources with a difference less than the threshold will be recruited into $\mathscr{A}$. Under the GLM setting, a natural loss function is the negative log-likelihood. For convenience, suppose n_0 is divisible by 3. According to (1), for any coefficient estimate $\boldsymbol{w}$. the average of negative log-likelihood on the r-th fold of target data $(\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})$ is

$$\hat{L}_0^{[r]}(\boldsymbol{w}) = -\frac{1}{n_0/3}\sum_{i=1}^{n_0/3}\log\rho(y_i^{(0)[r]}) - \frac{1}{n_0/3}(\boldsymbol{y}^{(0)[r]})^T\boldsymbol{X}^{(0)}\boldsymbol{w} + \frac{1}{n_0/3}\sum_{i=1}^{n_0/3}\psi(\boldsymbol{w}^T\boldsymbol{x}_i^{(0)[r]}). \tag{3}$$

The detailed algorithm is presented as Algorithm 2.

Algorithm 2: Trans-GLM

```
**Input:** target data $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$, all source data $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^K$, a constant $C_0 > 0$, penalty parameters $\{\{\lambda^{(k)[r]}\}_{k=0}^K\}_{r=1}^3$
**Output:** the estimated coefficient vector $\boldsymbol{\beta}$, and the determind transferring set $\mathscr{A}$
**Transferable source detection:** Randomly divide $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$ into three sets of equal size as $\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^3$
**for** $r = 1$ **to** 3 **do**
    $\boldsymbol{\beta}^{(0)[r]} \leftarrow$ fit the Lasso on $\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^3 \setminus (\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})$ with penalty parameter $\lambda^{(0)[r]}$
    $\boldsymbol{\beta}^{(k)[r]} \leftarrow$ run step 1 in Algorithm 1 with $(\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^3 \setminus (\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})) \cup (\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})$ and penalty parameter $\lambda^{(k)[r]}$ for all $k \neq 0$
    Calculate the loss function $\hat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)[r]})$ on $(\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})$ for $k = 1, \ldots, K$
**end**
$\hat{L}_0^{(k)} \leftarrow \sum_{r=1}^3 \hat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)[r]}) / 3$, $\hat{L}_0^{(0)} \leftarrow \sum_{r=1}^3 \hat{L}_0^{[r]}(\boldsymbol{\beta}^{(0)[r]}) / 3$, $\hat{\sigma} = \sqrt{\sum_{r=1}^3 (\hat{L}_0^{[r]}(\boldsymbol{\beta}^{(0)[r]}) - \hat{L}_0^{(0)})^2 / 2}$
$\mathscr{A} \leftarrow \{k \neq 0 : \hat{L}_0^{(k)} - \hat{L}_0^{(0)} \leq C_0(\hat{\sigma} \vee 0.01)\}$
$\mathscr{A}$ **– Trans – GLM:** $\boldsymbol{\beta} \leftarrow$ run Algorithm 1 using $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}k \in \{0\} \cup \mathscr{A}$
Output $\boldsymbol{\beta}$ and $\mathscr{A}$
```

It's important to point out that Algorithm 2 **does not** require the input of *h*. We will show that $\mathscr{A} = \mathscr{A}_h$ for some specific *h* if certain conditions hold, in Section 3.2. Furthermore, under these conditions, transferring with $\mathscr{A}$ will lead to a faster convergence rate compared to Lasso fitted on only the target data, when target sample size n_0 falls into some regime. This is the reason that this algorithm is called the *transferable* source detection algorithm.

2.5 Confidence intervals

In previous sections, we've discussed how to obtain a point estimator of the target coefficient vector $\boldsymbol{\beta}$ from the two-step transfer learning approach. In this section, we would like to construct the asymptotic confidence interval (CI) for each component of $\boldsymbol{\beta}$ based on that point estimate.

As described in the introduction, there have been quite a few works on high-dimensional GLM inference in the literature. In the following, we will propose a transfer learning procedure to construct CI based on the desparsified Lasso (Van de Geer et al., 2014). Recall that desparsified Lasso contains two main steps. The first step is to learn the inverse Fisher information matrix of GLM by nodewise regression (Meinshausen and Bühlmann, 2006). The second step is to “debias” the initial point estimator and then construct the asymptotic CI. Here, the estimator $\boldsymbol{\beta}$ from Algorithm 1 can be used as an initial point estimator. Intuitively, if the predictors from target and source data are similar and satisfy some sparsity conditions, it might be possible to use Algorithm 1 for learning the inverse Fisher information matrix of target data, which effectively combines the information from target and source data.

Before formalizing the procedure to construct the CI, let’s first define several additional notations. For any $\boldsymbol{w} \in \mathbb{R}^n$, denote $\boldsymbol{W}_{\boldsymbol{w}}^{(k)} = \mathrm{diag}\left(\sqrt{\psi''((\boldsymbol{x}_1^{(k)})^T \boldsymbol{w})}, \ldots, \sqrt{\psi''((\boldsymbol{x}_{n_k}^{(k)})^T \boldsymbol{w})}\right)$, $\boldsymbol{X}_{\boldsymbol{w}}^{(k)} = \boldsymbol{W}_{\boldsymbol{w}}^{(k)} \boldsymbol{X}^{(k)}$, $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(k)} = \mathbb{E}[\boldsymbol{x}^{(k)} (\boldsymbol{x}^{(k)})^T \psi''((\boldsymbol{x}^{(k)})^T \boldsymbol{w})]$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(k)} = n_k^{-1} (\boldsymbol{X}_{\boldsymbol{w}}^{(k)})^T \boldsymbol{X}_{\boldsymbol{w}}^{(k)}$. $\boldsymbol{X}_{\boldsymbol{w},j}^{(k)}$ represents the j-th column of $\boldsymbol{X}_{\boldsymbol{w}}^{(k)}$ and $\boldsymbol{X}_{\boldsymbol{w},-j}^{(k)}$ represents the matrix $\boldsymbol{X}_{\boldsymbol{w}}^{(k)}$ without the j-th column. $\boldsymbol{\Sigma}_{\boldsymbol{w},j,-j}^{(k)}$ represents the j-th row of $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(k)}$ without the diagonal (j, j) element, and $\boldsymbol{\Sigma}_{\boldsymbol{w},j,j}^{(k)}$ is the diagonal (j, j) element of $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{(k)}$.

Next, we explain the details of the CI construction procedure in Algorithm 3. In step 1, we obtain a point estimator $\boldsymbol{\beta}$ from $\mathscr{A}$ -Trans-GLM (Algorithm 1), given a specific transferring set $\mathscr{A}$. Then in steps 2-4, we estimate the target inverse Fisher information matrix $(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(0)})^{-1}$ as

$$\boldsymbol{\Theta} = \mathrm{diag}(\hat{\tau}_1^{-2}, \ldots, \hat{\tau}_p^{-2}) \begin{pmatrix} 1 & -\hat{\gamma}_{1,2}^{(0)} & \cdots & -\hat{\gamma}_{1,p}^{(0)} \\ -\hat{\gamma}_{2,1}^{(0)} & 1 & \cdots & -\hat{\gamma}_{2,p}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{\gamma}_{p,1}^{(0)} & -\hat{\gamma}_{p,2}^{(0)} & \cdots & 1 \end{pmatrix} \tag{4}$$

.

Finally in step 5, we “debias” $\boldsymbol{\beta}$ using the target data to get a new point estimator $\boldsymbol{b}$ which is asymptotically unbiased as

$$\boldsymbol{b} + \boldsymbol{\beta} + \frac{1}{n_0} \boldsymbol{\Theta} (\boldsymbol{X}^{(0)})^T [\boldsymbol{Y}^{(0)} - \boldsymbol{\psi}'(\boldsymbol{X}^{(0)} \boldsymbol{\beta})], \tag{5}$$

where $\boldsymbol{\psi}'(\boldsymbol{X}^{(0)} \boldsymbol{\beta}) := (\psi'((\boldsymbol{x}_1^{(0)})^T \boldsymbol{\beta}), \ldots, \psi'((\boldsymbol{x}_{n_0}^{(0)})^T \boldsymbol{\beta}))^T \in \mathbb{R}^{n_0}$.

It’s necessary to emphasize that the confidence level $(1 - \alpha)$ is for every single CI rather than for all p CIs simultaneously. As discussed in Sections 2.2 and 2.3 of Van de Geer et al. (2014), it is possible to get simultaneous CIs for different coefficient components and do multiple hypothesis tests when the design is fixed. In other cases, e.g., random design in different replications (which we focus on in this paper), multiple hypothesis testing might be more challenging.

Algorithm 3: Confidence interval construction via nodewise regression

```
Input: target data $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$, source data $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^{K}$, penalty parameters $\{\lambda_j\}_{j=1}^{p}$
    and $\{\widetilde{\lambda}_j\}_{j=1}^{p}$, transferring set $\mathscr{A}$, confidence level $(1-\alpha)$
Output: Level-$(1-\alpha)$ confidence interval $I_j$ for $\beta_j$ with $j = 1, \ldots, p$
Compute $\boldsymbol{\beta}$ via Algorithm 1
Compute $\widehat{\boldsymbol{\gamma}}_j^{\mathscr{A}} \leftarrow \arg\min_{\gamma}\left\{-\frac{1}{2(n_{\mathscr{A}}+n_0)}\sum k \in \{0\} \cup \mathscr{A} \, \|\boldsymbol{X}_{\beta,j}^{(k)} - \boldsymbol{X}_{\beta,-j}^{(k)}\boldsymbol{\gamma}\|_2^2 + \lambda_j \|\boldsymbol{\gamma}\|_1\right\}$ for $j = 1, \ldots, p$
Compute $\boldsymbol{\varrho}_j \leftarrow \arg\min_{\varrho}\left\{-\frac{1}{2n_0}\|\boldsymbol{X}_{\beta,j}^{(0)} - \boldsymbol{X}_{\beta,-j}^{(0)}(\widehat{\boldsymbol{\gamma}}_j^{\mathscr{A}} + \boldsymbol{\varrho})\|_2 + \widetilde{\lambda}_j \|\boldsymbol{\varrho}\|_1\right\}$
Compute $\widehat{\boldsymbol{\gamma}}_j^{(0)} \leftarrow \widehat{\boldsymbol{\gamma}}_j^{\mathscr{A}} + \boldsymbol{\varrho}_j, \boldsymbol{\Sigma}_\beta \leftarrow \sum_{k \in \{0\} \cup \mathscr{A}} \frac{n_k}{n_{\mathscr{A}}+n_0}\boldsymbol{\Sigma}_\beta^{(k)}, \widehat{\tau}_j^2 = \boldsymbol{\Sigma}_{\beta,j,j} - \boldsymbol{\Sigma}_{\beta,j,-j}\widehat{\boldsymbol{\gamma}}_j$ and calculate
    $\boldsymbol{\Theta}$ via (6), where $\widehat{\boldsymbol{\gamma}}_j^{(0)} = (\widehat{\gamma}_{j,1}^{(0)}, \ldots, \widehat{\gamma}_{j,j-1}^{(0)}, \widehat{\gamma}_{j,j+1}^{(0)}, \ldots, \widehat{\gamma}_{j,p}^{(0)})^T$.
Compute $I_j \leftarrow [\widehat{b}_j - \boldsymbol{\Theta}_j^T \boldsymbol{\Sigma}_\beta \boldsymbol{\Theta}_j q_{\alpha/2} / \sqrt{n_0}, \widehat{b}_j + \boldsymbol{\Theta}_j^T \boldsymbol{\Sigma}_\beta \boldsymbol{\Theta}_j q_{\alpha/2} / \sqrt{n_0}]$ for $j = 1, \ldots, p$, where
    $\widehat{b}_j$ is the $j$–th component of $\boldsymbol{b}$ in (7), and $q_{\alpha/2}$ is the $\alpha/2$ – left tail quantile of $N(0,1)$
Output $\{I_j\}_{j=1}^{p}$
```

3 Theory

In this section, we will establish theoretical guarantees on the three proposed algorithms. Section 3.1 provides a detailed analysis of Algorithm 1 with transferring set $\mathscr{A}_h$, which we denote as $\mathscr{A}_h$-Trans-GLM. Section 3.2 introduces certain conditions, under which we show that the transferring set $\mathscr{A}$ detected by Algorithm 2 (Trans-GLM) is equal to $\mathscr{A}_h$ for some h with high probability. Section 3.3 presents the analysis of Algorithm 3 with transferring set $\mathscr{A}_h$, where we prove a central limit theorem. For the proofs and some additional theoretical results, refer to supplementary materials.

3.1 Theory on $\mathscr{A}_h$-Trans-GLM

We first impose some common assumptions about GLM.

Assumption 1. *ψ is infinitely differentiable and strictly convex. We call a second-order differentiable function ψ strictly convex if* $\psi''(x) > 0$.

Assumption 2. *For any $\boldsymbol{a} \in \mathbb{R}^p$, $\boldsymbol{a}^T\boldsymbol{x}_i^{(k)}$, s are i.i.d. $\kappa_u \|\boldsymbol{a}\|_2^2$-subGaussian variables with zero mean for all $k = 0, \ldots, K$, where k_u is a positive constant. Denote the covariance matrix of $\boldsymbol{x}^{(k)}$ as $\Sigma^{(k)}$, with $\inf_k \lambda_{\min}(\boldsymbol{\Sigma}^{(k)}) \geq \kappa_l > 0$, where κ_l is a positive constant.*

Assumption 3. *At least one of the following assumptions hold: (M_ψ, U and $\overline{U}$ are some positive constants)*

- **i.** $\|\psi''\|_\infty \leq M_\psi < \infty$ *a.s.;*
- **ii.** $\sup_k \|\boldsymbol{x}^{(k)}\|_\infty \leq U < \infty$ *a.s.,* $\sup_k \sup_{|z| \leq \overline{U}} \psi''((\boldsymbol{x}^{(k)})^T\boldsymbol{w}^{(k)} + z) \leq M_\psi < \infty$ *a.s.*

Assumption 1 imposes the *strict convexity* and differentiability of ψ, which is satisfied by many popular distribution families, such as Gaussian, binomial, and Poisson distributions. Note that we do not require ψ to be *strongly convex* (that is, $\exists C > 0$, such that $\psi''(x) > C$), which relaxes Assumption 4 in Bastani (2021). It is easy to verify that ψ in logistic regression is in general not strongly convex with unbounded predictors. Assumption 2 requires the predictors in each source to be subGaussian with a well-behaved correlation structure. Assumption 3 is motivated by Assumption (GLM 2) in the full-length version of Negahban et al. (2009), which is imposed to restrict ψ'' in a bounded region in some sense. Note that linear regression and logistic regression satisfy condition (i), while Poisson regression with coordinate-wise bounded predictors and ℓ_1-bounded coefficients satisfies condition (ii).

Besides these common conditions on GLM, as discussed in Section 2.3, to guarantee the success of $\mathscr{A}_h$-Trans-GLM, we have to make sure that the estimator from the transferring step is close enough to $\boldsymbol{\beta}$. Therefore we introduce the following assumption, which guarantees $\boldsymbol{w}^{\mathscr{A}_h}$ defined in (2) with $\mathscr{A} = \mathscr{A}_h$ is close to $\boldsymbol{\beta}$.

Assumption 4. *Denote* $\boldsymbol{\Sigma}_h = \sum_{k \in \{0\} \cup \mathscr{A}_h} \alpha_k \mathbb{E}\left[\int_0^1 \psi''((\boldsymbol{x}^{(k)})^T \boldsymbol{\beta} + t(\boldsymbol{x}^{(k)})^T(\boldsymbol{w}^{\mathscr{A}_h} - \boldsymbol{\beta}))dt \cdot \boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T\right]$ *and* $\boldsymbol{\Sigma}_h^{(k)} = \mathbb{E}\left[\int_0^1 \psi''((\boldsymbol{x}^{(k)})^T \boldsymbol{\beta} + t(\boldsymbol{x}^{(k)})^T(\boldsymbol{w}^{(k)} - \boldsymbol{\beta}))dt \cdot \boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T\right]$. *It holds that*

$$\sup_{k \in \{0\} \cup \mathscr{A}_h} \|\boldsymbol{\Sigma}_h^{-1} \boldsymbol{\Sigma}_h^{(k)}\|_1 < \infty.$$

Remark 1. *A sufficient condition for Assumption 4 to hold is* $(\boldsymbol{\Sigma}^{(k)}_{\boldsymbol{w}^{\mathscr{A}_h}, \boldsymbol{\beta}})^{-1} \boldsymbol{\Sigma}^{(k')}_{\boldsymbol{w}^{(k')}, \boldsymbol{\beta}}$ *has positive diagonal elements and is diagonally dominant, for any* $k \neq k'$ *in* $\mathscr{A}_h$, *where* $\boldsymbol{\Sigma}^{(k)}_{\boldsymbol{w}, \boldsymbol{\beta}} := \mathbb{E}\left[\int_0^1 \psi''((\boldsymbol{x}^{(k)})^T \boldsymbol{\beta} + t(\boldsymbol{x}^{(k)})^T(\boldsymbol{w} - \boldsymbol{\beta}))dt \cdot \boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T\right]$ *for any* $\boldsymbol{w} \in \mathbb{R}^p$.

In the linear case, this assumption can be further simplified as a restriction on heterogeneity between target predictors and source predictors. More discussions can be found in Condition 4 of Li et al. (2021). Now, we are ready to present the following main result for the $\mathscr{A}_h$-Trans-GLM algorithm. Define the parameter space as

$$\Xi(s, h) = \left\{\boldsymbol{\beta}, \{\boldsymbol{w}^{(k)}\}_{k \in \mathscr{A}_h} : \|\boldsymbol{\beta}\|_0 \leq s, \sup_{k \in \mathscr{A}_h} \|\boldsymbol{w}^{(k)} - \boldsymbol{\beta}\|_1 \leq h\right\}.$$

Given s and h, we compress parameters $\boldsymbol{\beta}$, $\{\boldsymbol{w}^{(k)}\}_{k \in \mathscr{A}_h}$ into a parameter set $\boldsymbol{\xi}$ for simplicity.

Theorem 1 (ℓ_1/ℓ_2-estimation error bound of $\mathscr{A}_h$-Trans-GLM with Assumption 4).

Assume Assumptions 1, 2 and 4 hold. Suppose $h \ll \sqrt{\frac{n_0}{\log p}}$, $h \leq C\sqrt{s}$, $n_0 \geq C \log p$ *and* $n_{A_h} \geq Cs \log p$, *where* $C > 0$ *is a constant. Also assume Assumption 3.(i) holds or Assumption 3.(ii) with* $h \leq C'U^{-1}\overline{U}$ *for some* $C' > 0$ *holds. We take* $\lambda_w = C_w \sqrt{\frac{\log p}{n_{\mathscr{A}_h} + n_0}}$ *and* $\lambda_\delta = C_\delta \sqrt{\frac{\log p}{n_0}}$, *where* C_W *and* C_δ *are sufficiently large positive constants. Then*

$$\sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_2 \lesssim \left(\frac{s \log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/2} + \left[\left(\frac{\log p}{n_0}\right)^{1/4} h^{1/2}\right] \wedge h\right) \geq 1 - n_0^{-1}, \tag{6}$$

$$\sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_1 \lesssim s\left(\frac{\log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/2} + h\right) \geq 1 - n_0^{-1}. \tag{7}$$

Remark 2. *When* $h \ll s\sqrt{\frac{\log p}{n_0}}$, $n_{\mathscr{A}_h} \gg n_0$, *the upper bounds in* (6) *and* (7) *are better than the classical Lasso* ℓ_2*-bound* $O_p\left(\sqrt{\frac{s \log p}{n_0}}\right)$ *and* ℓ_1*-bound* $O_p\left(s\sqrt{\frac{\log p}{n_0}}\right)$ *using only target data.*

Similar to Theorem 2 in Li et al. (2021), we can show the following lower bound of ℓ_1/ℓ_2-estimation error in regime $\Xi(s, h)$ in the minimax sense.

Theorem 2 (ℓ_1/ℓ_2-minimax estimation error bound). *Assume Assumptions 1, 2 and 4 hold. Also assume Assumption 3.(i) holds or Assumption 3.(ii) with* $n_0 \gtrsim s^2 \log p$. *Then*

$$\inf_{\boldsymbol{\beta}} \sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_2 \gtrsim \left(\frac{s \log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/2} + \left(\frac{s \log p}{n_0}\right)^{1/2} \wedge \left[\left(\frac{\log p}{n_0}\right)^{1/4} h^{1/2}\right] \wedge h\right) \geq \frac{1}{2},$$
$$\inf_{\boldsymbol{\beta}} \sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_1 \gtrsim s\left(\frac{\log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/2} + \left[s\left(\frac{\log p}{n_0}\right)^{1/2}\right] \wedge h\right) \geq \frac{1}{2}.$$

Remark 3. Theorem 2 *indicates that under conditions on h required by Theorem 1 (*$h \lesssim s\sqrt{\log p / n_0}$), $\mathscr{A}_h$*-Trans-GLM achieves the minimax optimal rate of* ℓ_1/ℓ_2*-estimation error bound.*

Next, we present a similar upper bound, which is weaker than the bound above but holds without requiring Assumption 4.

Theorem 3 (ℓ_1/ℓ_2-estimation error bound of $\mathscr{A}_h$-Trans-GLM without Assumption 4). *Assume Assumptions 1 and 2 hold. Suppose* $h \ll \sqrt{\frac{n_0}{\log p}}$, $h \leq C s^{-1/2}$, $n_0 \geq C \log p$ *and* $n_{A_h} \geq Cs$ log p, *where* $C > 0$ *is a constant. Also assume Assumption 3.(i) holds or Assumption 3.(ii) with* $h \leq C' U^{-1} \bar{U}$ *for some* $C' > 0$ *holds. We take* $\lambda_w = C_w\left(\sqrt{\frac{\log p}{n_{\mathscr{A}_h} + n_0}} + h\right)$ *and* $\lambda_\delta = C_\delta\sqrt{\frac{\log p}{n_0}}$, *where* C_w *and* C_δ *are sufficiently large positive constants. Then*

$$\sup_{\boldsymbol{\xi} \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_2 \lesssim \left(\frac{s \log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/2} + \sqrt{s}h + \left[\left(\frac{\log p}{n_0}\right)^{1/4} h^{1/2}\right] \wedge h\right) \geq 1 - n_0^{-1},$$
$$\sup_{\boldsymbol{\xi} \in \Xi(s,h)} \mathbb{P}\left(\| \boldsymbol{\beta} - \boldsymbol{\beta} \|_1 \lesssim s\sqrt{\frac{\log p}{n_{\mathscr{A}_h} + n_0}} + sh\right) \geq 1 - n_0^{-1}.$$

Remark 4. *When* $h \ll \sqrt{\frac{\log p}{n_0}}$ *and* $n_{A_h} \gg n_0$*, the upper bounds in (i) and (ii) are better than the classical Lasso bound* $O_p\left(\sqrt{\frac{\log p}{n_0}}\right)$ *with target data.*

Comparing the results in Theorems 1 and 3, we know that with Assumption 4, we could get sharper ℓ_1/ℓ_2-estimation error bounds.

3.2 Theory on the transferable source detection algorithm

In this section, we will show that under certain conditions, our transferable set detection algorithm (Trans-GLM) can recover the level-h transferring set $\mathscr{A}_h$ for some specific h, that is, $\mathscr{A} = \mathscr{A}_h$ with high probability. Under these conditions, transferring with $\mathscr{A}$ will lead to a faster convergence rate compared to Lasso fitted on the target data, when the target sample size n_0 falls into certain regime. But as we described in Section 2.4, Algorithm 2 does not require any explicit input of h.

The corresponding population version of $\widehat{L}_0^{[r]}(\boldsymbol{w})$ defined in (3) is

$$\begin{aligned} L_0(\boldsymbol{w}) &= -\mathbb{E}[\log \rho(y^{(0)})] - \mathbb{E}[y^{(0)}\boldsymbol{w}^T\boldsymbol{x}^{(0)}] + \mathbb{E}[\psi(\boldsymbol{w}^T\boldsymbol{x}^{(0)})] \\ &= -\mathbb{E}[\log \rho(y^{(0)})] - \mathbb{E}[\psi'(\boldsymbol{w}^T\boldsymbol{x}^{(0)})\boldsymbol{w}^T\boldsymbol{x}^{(0)}] + \mathbb{E}[\psi(\boldsymbol{w}^T\boldsymbol{x}^{(0)})]\,. \end{aligned}$$

Based on Assumption 6, similar to (2), for $\{k\}$-Trans-GLM (Algorithm 1 with $\mathscr{A} = \{k\}$) used in Algorithm 2, consider the following population version of estimators from the transferring step with respect to target data and the k-th source data, which is the solution $\boldsymbol{\beta}^{(k)}$ of equation $\sum_{j \in \{0,k\}} \alpha_j^{(k)} \mathbb{E}\left\{[\psi'((\boldsymbol{\beta}^{(k)})^T\boldsymbol{x}^{(k)}) - \psi'((\boldsymbol{w}^{(k)})^T\boldsymbol{x}^{(k)})]\boldsymbol{x}^{(k)}\right\} = 0$, where $\alpha_0^{(k)} = \frac{2n_0/3}{2n_0/3 + n_k}$ and $\alpha_k^{(k)} = \frac{n_k}{2n_0/3 + n_k}$. Define $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}$. Next, let's impose a general assumption to ensure the identifiability of some $\mathscr{A}_h$ by Trans-GLM.

Assumption 5 (Identifiability of $\mathscr{A}_h$). *Denote* $\mathscr{A}_h^c = \{1, \ldots, K\} \setminus \mathscr{A}_h$*. Suppose for some h, we have*

$$\mathbb{P}\left(\sup_r \left| \widehat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)[r]}) - \widehat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)}) \right| > \Upsilon_1^{(k)} + \zeta\Gamma_1^{(k)}\right) \lesssim g_1^{(k)}(\zeta),$$
$$\mathbb{P}\left(\sup_r \left| \widehat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)}) - L_0(\boldsymbol{\beta}^{(k)}) \right| > \zeta\Gamma_2^{(k)}\right) \lesssim g_2^{(k)}(\zeta),$$

where $g_1^{(k)}(\zeta), g_2^{(k)}(\zeta) \to 0$ *as* $\zeta \to \infty$*. Assume* $\inf_{k \in \mathscr{A}_h^c} \lambda_{\min}(\mathbb{E}[\int_0^1 \psi''((1-t)(\boldsymbol{x}^{(0)})^T\boldsymbol{\beta} + t(\boldsymbol{x}^{(0)})^T\boldsymbol{\beta}^{(k)})dt \cdot \boldsymbol{x}^{(0)}(\boldsymbol{x}^{(0)})^T]) := \underline{\lambda} > 0$*, and*

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}\|_2 \geq \underline{\lambda}^{-1/2}\left[C_1\left(\sqrt{\Gamma_1^{(0)}} \vee \sqrt{\Gamma_2^{(0)}} \vee 1\right) + \sqrt{2\Upsilon_1^{(k)}}\right], \ \forall k \in \mathscr{A}_h^c \tag{8}$$

$$\Upsilon_1^{(k)} + \Gamma_1^{(k)} + \Gamma_2^{(k)} + h^2 = O(1),\ \forall\, k \in \mathscr{A}_h;\ \ \Gamma_1^{(k)} = O(1), \Gamma_2^{(k)} = O(1),\ \forall\, k \in \mathscr{A}_h^c, \tag{9}$$

where $C_1 > 0$ is sufficiently large.

Remark 5. *Here we use generic notations $\Upsilon_1^{(k)}$, $\Gamma_1^{(k)}$, $\Gamma_2^{(k)}$, $g_1^{(k)}(\zeta)$ and $g_2^{(k)}(\zeta)$. We show their explicit forms under linear, logistic, and Poisson regression models in Proposition 1 in* Section S. 1.2.1 of supplements.

Remark 6. *Condition* (8) *guarantees that for the sources not in $\mathscr{A}_h$, there is a sufficiently large gap between the population-level coefficient from the transferring step and the true coefficient of target data. Condition* (9) *guarantees the variations of* $\sup_r \mid \hat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)[r]}) - \hat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)}) \mid$ *and* $\sup_r \mid \hat{L}_0^{[r]}(\boldsymbol{\beta}^{(k)}) - L_0(\boldsymbol{\beta}^{(k)}) \mid$ *are shrinking as the sample sizes go to infinity.*

Based on Assumption 5, we have the following detection consistency property.

Theorem 4 (Detection consistency of $\mathscr{A}_h$). *For Algorithm 2 (Trans-GLM), with Assumption 5 satisfied for some h, for any $\delta > 0$ there exist constants $C'(\delta)$ and $N = N(\delta) > 0$ such that when $C_0 = C'(\delta)$ and* $\mathrm{m\ in}_{k \in \{0\} \cup \mathscr{A}_h} n_k > N(\delta)$, $\mathbb{P}(^{\mathscr{A}} = \mathscr{A}_h) \geq 1 - \delta$.

Then Algorithm 2 has the same high-probability upper bounds of ℓ_1/ℓ_2-estimation error as those in Theorems 1 and 3 under the same conditions, respectively.

Remark 7. *We would like to emphasize again that Algorithm 2 does not require the explicit input of h. Theorem 4 tells us that the transferring set $\mathscr{A}$ suggested by Trans-GLM will be $\mathscr{A}_h$ for some h, under certain conditions.*

Next, we attempt to provide a sufficient and more explicit condition (Corollary 1) to ensure that Assumption 5 hold. Recalling the procedure of Algorithm 2, we note that it relies on using the negative log-likelihood as the similarity metric between target and source data, where the accurate estimation of coefficients or log-likelihood for GLM under the high-dimensional setting depends on the sparse structure. Therefore, in order to provide an explicit and sufficient condition for Assumption 6 to hold, we now impose a “weak” sparsity assumption on both $\boldsymbol{w}^{(k)}$ and $\boldsymbol{\beta}^{(k)}$ with $k \in \mathscr{A}_h^c$ for some h. Note that the source data in $\mathscr{A}_h$ automatically satisfy the sparsity condition due to the definition of $\mathscr{A}_h$.

Assumption 6. *For some h and any $k \in \mathscr{A}_h^c$, we assume $\boldsymbol{w}^{(k)}$ and $\boldsymbol{\beta}^{(k)}$ can be decomposed as follows with some s' and $h' > 0$:*

- **i.** $\boldsymbol{w}(k) = \boldsymbol{\varsigma}(k) + \boldsymbol{\vartheta}(k)$, *where* $\|\boldsymbol{\varsigma}(k)\|_0 \leq s'$ *and* $\|\boldsymbol{\vartheta}(k)\|_1 \leq h'$;
- **ii.** $\boldsymbol{\beta}(k) = \boldsymbol{l}(k) + \boldsymbol{\varpi}(k)$, *where* $\|\boldsymbol{l}(k)\|_0 \leq s'$ *and* $\|\boldsymbol{\varpi}(k)\|_1 \leq h'$.

Corollary 1. *Assume Assumptions 1, 2, 6 and* $\inf_{k \in \mathscr{A}_h^c} \lambda_{\min}\left(\mathbb{E}[\int_0^1 \psi''((1-t)(\boldsymbol{x}^{(0)})^T\boldsymbol{\beta} + t(\boldsymbol{x}^{(0)})^T\boldsymbol{\beta}^{(k)})dt \cdot \boldsymbol{x}^{(0)}(\boldsymbol{x}^{(0)})^T]\right) := \underline{\lambda} > 0$ *hold. Also assume* $\sup_{k \in \mathscr{A}_h^c} \|\boldsymbol{\beta}^{(k)}\|_\infty < \infty$, $\sup_k \|\boldsymbol{w}^{(k)}\|_\infty < \infty$. *Let* $\lambda^{(k)[r]} = C\left(\sqrt{\frac{\log p}{n_k + n_0}} + h\right)$ *when* $k \in \mathscr{A}_h$, $\lambda^{(k)[r]} = C\sqrt{\frac{\log p}{n_k + n_0}} \cdot (1 \vee \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}\|_2 \vee \|\boldsymbol{w}^{(k)} - \boldsymbol{\beta}\|_2)$ *when* $k \in \mathscr{A}_h^c$ *and* $\lambda^{(0)} = C\sqrt{\frac{\log p}{n_0}}$ *for some sufficiently large constant C > 0. Then we have the following sufficient conditions to make Assumption 5 hold for logistic, linear and Poisson regression models. Denote*

$$\Omega = \sqrt{h'}\left(\frac{\log p}{\min_{k \in \mathscr{A}_h} n_k + n_0}\right)^{1/4} + \left(\frac{s' \log p}{\min_{k \in \mathscr{A}_h} n_k + n_0}\right)^{1/4}[(s \vee s')^{1/4} + \sqrt{h'}] + \left(\frac{\log p}{\min_{k \in \mathscr{A}_h} n_k + n_0}\right)^{1/8}(h')^{1/4}[(s \vee s')^{1/8} + (h')^{1/4}]$$

i. For logistic regression models, we require

$$\inf_{k \in \mathscr{A}_h} n_k \gg s \log p, \quad n_0 \gg \left\{[s \vee s' + (h')^2] \vee \Omega^2\right\} \cdot \log K,$$
$$\inf_{k \in \mathscr{A}_h^c} \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}\|_2 \gtrsim \left(\frac{s \log p}{n_0}\right)^{1/4} \vee 1 + \Omega, \quad h \ll s^{-1/2}.$$

ii. For linear models, we require

$$\inf_{k \in \mathscr{A}_h} n_k \gg s^2 \log p, \quad n_0 \gg \left\{[(s \vee s')^2 + (h')^4] \vee [(s \vee s' + (h')^2)\Omega^2]\right\} \cdot \log K,$$
$$\inf_{k \in \mathscr{A}_h^c} \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}\|_2 \gtrsim \left(\frac{s^2 \log p}{n_0}\right)^{1/4} \vee 1 + \left[(s')^{1/4} + \sqrt{h'}\right]\Omega, \quad h \ll s^{-1}.$$

iii. For Poisson regression models, we require

$$\inf_{k \in \mathscr{A}_h} n_k \gg s^2 \log p, \quad n_0 \gg \left[(s \vee s' + h') \vee \Omega^2\right] \cdot \log K, \quad U(s \vee s' + h \vee h') \lesssim 1,$$
$$\inf_{k \in \mathscr{A}_h^c} \|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}\|_2 \gtrsim \left(\frac{s \log p}{n_0}\right)^{1/4} \vee 1 + \left[(s')^{1/4} + \sqrt{h'}\right]\Omega, \quad h \ll s^{-1}.$$

Under Assumptions 1, 2, and the sufficient conditions derived in Corollary 1, by Theorem 4, we can conclude that $\mathscr{A} = \mathscr{A}_h$ for some *h*. Note that we don't impose Assumption 4 here. Remark 4 indicates that, for $\mathscr{A}_h$-Trans-GLM to have a faster convergence rate than Lasso on target data, we need $h \ll \sqrt{\frac{\log p}{n_0}}$ and $n_{A_h} \gg n_0$. Suppose $s' \asymp s$, $h' \lesssim s^{1/2}$. Then for logistic regression models, when $s \log K \ll n_0 \ll s \log p$, the sufficient condition implie $h \ll s^{-1/2} \ll \sqrt{\frac{\log p}{n_0}}$. For linear models, when $s^2 \log K \ll n_0 \ll s^2 \log p$, $h \ll s^{-1} \ll \sqrt{\frac{\log p}{n_0}}$. And for Poisson models, when

$s \log K \ll n_0 \ll s^2 \log p$, $h \ll s^{-1} \ll \sqrt{\frac{\log p}{n_0}}$. This implies that when target sample size n_0 is within certain regimes and there are many more source data points than target data points, Trans-GLM can lead to a better ℓ_2-estimation error bound than the classical Lasso on target data.

3.3 Theory on confidence interval construction

In this section, we will establish the theory for our confidence interval construction procedure described in Algorithm 3. First, we would like to review and introduce some notations. In Section 2.5, we defined $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(k)} = \mathbb{E}[\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^T \psi''((\boldsymbol{x}^{(k)})^T \boldsymbol{\beta})]$. Let $\boldsymbol{\Theta} = (\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(0)})^{-1}$ and $K_{A_h} = |\mathscr{A}_h|$. Define

$$\boldsymbol{\gamma}_j^{(k)} = \underset{\boldsymbol{\gamma} \in \mathbb{R}^{p-1}}{\arg\min} \mathbb{E}\left\{\psi''(\boldsymbol{\beta}^T \boldsymbol{x}^{(k)}) \cdot [\boldsymbol{x}_j^{(k)} - (\boldsymbol{x}_{-j}^{(k)})^T \boldsymbol{\gamma}]^2\right\} = (\boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(k)})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(k)},$$

which is closely related to $(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(k)})^{-1}$ and $\boldsymbol{\gamma}_j^{(0)}$ can be viewed as the population version of $\hat{\boldsymbol{\gamma}}_j^{(0)}$. And $\boldsymbol{\Sigma}_{\boldsymbol{\beta}, j, -j}^{(k)}$ represents the j-th row without the (j, j) diagonal element of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(k)}$. $\boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(k)}$ denotes the submatrix of $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{(k)}$ without the j-th row and j-th column. Suppose

$$\sup_{k \in \mathscr{A}_h, \, j = 1:p} \| (\boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(0)})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, j}^{(0)} - (\boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(k)})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\beta}, -j, -j}^{(k)} \|_1 \leq h_1,$$
$$\sup_{k \in \mathscr{A}_h, \, j = 1:p} \left[| \boldsymbol{\Sigma}_{\boldsymbol{\beta}, j, j}^{(k)} - \boldsymbol{\Sigma}_{\boldsymbol{\beta}, j, j}^{(0)} | \vee | (\boldsymbol{\Sigma}_{\boldsymbol{\beta}, j, -j}^{(k)} - \boldsymbol{\Sigma}_{\boldsymbol{\beta}, j, -j}^{(0)}) \boldsymbol{\gamma}_j^{(0)} | \right] \leq h_{\max}.$$

Then by the definition of $\boldsymbol{\gamma}_j^{(k)}$,

$$\sup_{k \in \mathscr{A}_h, \, j = 1:p} \| \boldsymbol{\gamma}_j^{(k)} - \boldsymbol{\gamma}_j^{(0)} \|_1 \lesssim h_1,$$

which is similar to our previous setting $\sup_{k \in \mathscr{A}_h} \| \boldsymbol{w}^{(k)} - \boldsymbol{\beta} \|_1 \leq h$. This motivates us to apply a similar two-step transfer learning procedure (steps 2-4 in Algorithm 3) to learn $\boldsymbol{\gamma}_j^{(0)}$ for $j = 1, \ldots, p$. We impose the following set of conditions.

Assumption 7. *Suppose the following conditions hold:*

i. $\sup_{k \in \{0\} \cup \mathscr{A}_h} \| \boldsymbol{x}^{(k)} \|_\infty \leq U < \infty, \quad \sup_{k \in \{0\} \cup \mathscr{A}_h} | (\boldsymbol{x}^{(k)})^T \boldsymbol{w}^{(k)} | \leq U' < \infty$ *a.s.;*

ii. $\sup_j \| \boldsymbol{\gamma}_j^{(0)} \|_0 / s < \infty, \quad \sup_{j \in 1:p, k \in \{0\} \cup \mathscr{A}_h} | (\boldsymbol{x}^{(k)})^T \boldsymbol{\gamma}_j^{(0)} | \leq U'' < \infty$ *a.s.;*

iii. $\inf_{k \in \{0\} \cup \mathscr{A}_h} \lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{w}^{(k)}}^{(k)}) \geq \underline{U} > 0;$

iv. $\sup_{k \in \{0\} \cup \mathscr{A}_h} \sup_{|z| \leq \bar{U}} \psi'''((\boldsymbol{x}^{(k)})^T \boldsymbol{w}^{(k)} + z) \leq M_\psi < \infty$ *a.s.*

v. $\sup_{k \in \{0\} \cup \mathscr{A}_h} \| (\boldsymbol{\Sigma}_{\beta, -j, -j}^{\mathscr{A}_h})^{-1} \boldsymbol{\Sigma}_{\beta, -j, -j}^{(k)} \|_1 < \infty$, *where* $\boldsymbol{\Sigma}_{\beta}^{\mathscr{A}_h} = \sum_{k \in \{0\} \cup \mathscr{A}_h} \alpha_k \boldsymbol{\Sigma}_{\beta}^{(k)}$;

vi. $\min_{k \in \mathscr{A}_h} n_k \gtrsim n_0,\ n_0 \gg \frac{s^3 (\log p)^2}{K_{\mathscr{A}_h}^2} \vee K_{\mathscr{A}_h},\ n_{\mathscr{A}_h} + n_0 \gg s^2 \log p$;

vii. $h_1 \lesssim s^{-1/2} \wedge \left[\sqrt{\frac{n_0}{\log p}}\left(\frac{\sqrt{K_{\mathscr{A}_h}}}{s} \wedge 1\right)\right],\ h_1 \vee h \ll \frac{K_{\mathscr{A}_h} n_0^{1/2}}{s^2 (\log p)^{3/2}} \wedge \frac{n_0^{1/4}}{s^{1/2} (\log p)^{1/4}},\ h h_1^{1/2}.$
$\ll$
$n_0^{-1/4} (\log p)^{-1/4} \left(\frac{K_{\mathscr{A}_h}}{s} \wedge 1\right),\ h^{5/2} h_1 \ll n_0^{-3/4} (s \log p)^{-1/4},\ h_1 \ll \frac{K_{\mathscr{A}_h}^{1/2} n_0^{1/2}}{s^{3/2} (\log p)^{1/2}} \wedge$
$\frac{K_{\mathscr{A}_h}^{3/2} n_0^{1/2}}{s^{5/2} (\log p)^{3/2}}$,
$h_1 h^{1/2} \ll \frac{n_0^{1/4}}{s (\log p)^{1/4}} \wedge \frac{K_{\mathscr{A}_h} n_0^{1/4}}{s^2 (\log p)^{5/4}},\ h \ll \frac{K_{\mathscr{A}_h}^{1/2}}{(s \log p)^{1/2}} \wedge \frac{1}{n_0^{1/4} (\log p)^{1/2}},\ h_{\max} \ll$
$s^{-1/2} \wedge \left(\frac{1}{s}\sqrt{\frac{K_{\mathscr{A}_h}}{\log p}}\right)$,
$h h_{\max} \ll n_0^{-1/2}$

Remark 8. *Conditions (i)-(iii) are motivated from conditions of Theorem 3.3 in* Van de Geer et al. (2014). *Note that in* Van de Geer et al. (2014), *they define* $s_j = \| \boldsymbol{\gamma}_j^{(0)} \|_0$ *and treat* s_j *and s as two different parameters. Here we require* $\sup_j s_j \lesssim s$ *just for simplicity (otherwise condition (vii) would be more complicated). Condition (iv) requires the inverse link function to behave well, which is similar to Assumption 3. Condition (v) is similar to Assumption 4 to guarantee the success of the two-step transfer learning procedure to learn* $\boldsymbol{\gamma}^{(0)}$ *in Algorithm 3 with a fast rate. Without condition (v), the conclusions in the following Theorem 5 may still hold but under a stronger condition on h,* h_1 *and* $h_{\max}$, *and the rate (34) may be worse. We do not explore the details in this paper and leave them to interested readers. Conditions (vi) and (vii) require that the sample size is sufficiently large and the distance between target and source is not too large. In condition (vi),* $\min_{k \in \mathscr{A}_h} n_k \gtrsim n_0$ *is not necessary and the only reason we add it here is to simplify condition (vii).*

Remark 9. *When* $\mathbf{x}^{(k)}$*'s are from the same distribution,* $h_1 = h_{\max} = 0$. *In this case, we can drop the debiasing step to estimate* $\hat{\boldsymbol{\gamma}}_j^{(0)}$ *in Algorithm 3 as well as condition (v). Furthermore, condition (vii) can be significantly simplified and only* $h \ll \frac{K_{\mathscr{A}_h} n_0^{1/2}}{s^2 (\log p)^{3/2}} \wedge \frac{n_0^{1/4}}{s^{1/2} (\log p)^{1/4}} \wedge \frac{K_{\mathscr{A}_h}^{1/2}}{(s \log p)^{1/2}} \wedge \frac{1}{n_0^{1/4} (\log p)^{1/2}}$ *is needed.*

Remark 10. *From conditions (vi) and (vii), we can see that as long as* $K_{A_h} \lesssim s(\log p)^{2/3}$, *the conditions become milder as* K_{A_h} *increases.*

Now, we are ready to present our main result for Algorithm 3.

Theorem 5. *Under Assumptions 1-4 and Assumption 7,*

$$\frac{\sqrt{n_0}(\hat{b}_j - \beta_j)}{\sqrt{\boldsymbol{\Theta}_j^T \boldsymbol{\Sigma}_\beta \boldsymbol{\Theta}_j}} - \xrightarrow{d} N(0, 1), \tag{10}$$

and

$$| \boldsymbol{\Theta}_j^T \boldsymbol{\Sigma}_\beta \boldsymbol{\Theta}_j - \boldsymbol{\Theta}_{jj} | \lesssim s\sqrt{\frac{\log p}{n_{\mathscr{A}_h} + n_0}} + \sqrt{s}\left[h^{1/2}\left(\frac{\log p}{n_0}\right)^{1/4} \wedge h\right] + (s h_1)^{1/2}\left(\frac{\log p}{n_0}\right)^{1/4} + (s h_1)^{1/2}\left[\left(\frac{s \log p}{n_{\mathscr{A}_h} + n_0}\right)^{1/4} + \left(h^{1/4}\left(\frac{\log p}{n_0}\right)^{1/8}\right) \wedge h^{1/2}\right] + \sqrt{s}\, h_{max}, \tag{11}$$

for $j = 1, \ldots, p$, *with probability at least* $1 - K_{\mathscr{A}_h} n_0^{-1}$.

Theorem 5 guarantees that under certain conditions, the $(1 - \alpha)$-confidence interval for each coefficient component obtained in Algorithm 3 has approximately level $(1 - \alpha)$ when the sample size is large. Also, if we compare the rate of (34) with the rate $O_p(s\sqrt{\log p / n_0})$ in Van de Geer et al. (2014) (see the proof of Theorem 3.1), we can see that when $h \ll s\sqrt{\frac{\log p}{n_0}}$, $h_1 \ll \sqrt{\frac{s \log p}{n_0}} \cdot \left[s^{1/2} \wedge \left(\frac{n_{\mathscr{A}_h} + n_0}{n_0}\right)^{1/4}\right]$, $h_1^{1/2} h^{1/4} \ll s^{1/2}\left(\frac{\log p}{n_0}\right)^{3/8}$ and $h_{\max} \ll \sqrt{\frac{s \log p}{n_0}}$, the rate is better than that of desparsified Lasso using only target data.

4 Numerical Experiments

In this section, we demonstrate the power of our GLM transfer learning algorithms via extensive simulation studies and a real-data application. In the simulation part, we study the performance of different methods under various settings of h. The methods include *Trans-GLM* (Algorithm 2), *naïve-Lasso* (Lasso on target data), $\mathscr{A}_h$-*Trans-GLM* (Algorithm 1 with $\mathscr{A} = \mathscr{A}_h$) and *Pooled-Trans-GLM* (Algorithm 1 with all sources). In the real-data study, besides naïve-Lasso, Pooled-Trans-GLM, and Trans-GLM, additional methods are explored for comparison, including support vector machines (SVM), decision trees (Tree), random forests (RF) and Adaboost algorithm with trees (Boosting). We run these benchmark methods twice. First, we fit the models on only the target data, then at the second time, we fit them a combined data of target and all sources, which is called a pooled version. We use the original method name to denote the corresponding method implemented on target data, and add a prefix “Pooled” to denote the corresponding method implemented on target and all source data. For example, Pooled-SVM represents SVM fitted on all data from target and sources.

All experiments are conducted in R. We implement our GLM transfer learning algorithms in a new R package `glmtrans`, which is available on CRAN. More implementation details can be found in Section S. 1.3.1 in the supplements.

4.1 Simulations

4.1.1 Transfer learning on $\mathscr{A}_h$—In this section, we study the performance of $\mathscr{A}_h$-Trans-GLM and compare it with that of naïve-Lasso. The purpose of the simulation is to verify that $\mathscr{A}_h$-Trans-GLM can outperform naïve-Lasso in terms of the target coefficient estimation error, when h is not too large.

Consider the simulation setting as follows. We set the target sample size $n_0 = 200$ and source sample sample size $n_k = 100$ for each $k \neq 0$. The dimension $p = 500$ for both target and source data. For the target, the coefficient is set to be $\boldsymbol{\beta} = (0.5 \cdot \mathbf{1}_s, \mathbf{0}_{p-s})^T$, where $\mathbf{1}_s$ has all s elements 1 and $\mathbf{0}_{p-s}$ has all $(p - s)$ elements 0, where s is set to be 5. Denote $R\, p^{(k)}$ as p independent Rademacher variables (being −1 or 1 with equal probability) for any k. $\boldsymbol{R}p^{(k)}$ is independent with $\boldsymbol{R}p^{(k')}$ for any $k \neq k2$. For any source data k in $\mathscr{A}_h$, we set $\boldsymbol{w}^{(k)} = \boldsymbol{\beta} + (h\ /\ p)^{\boldsymbol{R}_p^{(k)}}$. For linear and logistic regression models, predictors from target $\boldsymbol{x}_i^{(0)} \overset{i.i.d.}{\sim} N(\mathbf{0}_p, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = [\Sigma_{jj2}]_{p \times p}$ where $\Sigma_{jj2} = 0.5^{|j-j'|}$, for all $i = 1, \ldots, n$. And for $k \in \mathscr{A}_h$, we generate p-dimensional predictors from $N(\mathbf{0}_p, \boldsymbol{\Sigma} + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T)$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}_p, 0.3^2 \boldsymbol{I}_p)$ and is independently generated. For Poisson regression model, predictors are from the same Gaussian distributions as those in linear and binomial cases with coordinate-wise truncation at ±0.5.

Note that naïve-Lasso is fitted on only target data, and $\mathscr{A}_h$-Trans-GLM denotes Algorithm 1 on source data in $\mathscr{A}_h$ as well as target data. We train naïve-Lasso and $\mathscr{A}_h$-Trans-GLM models under different settings of h and $K_{\mathscr{A}_h}$, then calculate the ℓ_2-estimation error of $\boldsymbol{\beta}$. All the experiments are replicated 200 times and the average ℓ_2-estimation errors of $\mathscr{A}_h$-Trans-GLM and naïve-Lasso under linear, logistic, and Poisson regression models are shown in Figure 1.

From Figure 1, it can be seen that $\mathscr{A}_h$-Trans-GLM outperforms naïve-Lasso for most combinations of h and K. As more and more source data become available, the performance of $\mathscr{A}_h$-Trans-GLM improves significantly. This matches our theoretical analysis because the ℓ_2-estimation error bounds in Theorems 1 and 3 become sharper as $n_{\mathscr{A}_h}$ grows. When h increases, the performance of $\mathscr{A}_h$-Trans-GLM becomes worse.

We also apply the inference algorithm 3 with $\mathscr{A}_h$ and compare it with desparsified Lasso (Van de Geer et al., 2014) on only target data. Recall the notations we used in Section 3.3. Here we consider 95% confidence intervals (CIs) for each component of coefficient $\boldsymbol{\beta}$, and report three evaluation metrics in Figure 2 when $h = 20$ under different $K_{\mathscr{A}_h}$: (i) the average of estimation error of $\boldsymbol{\Theta}_{jj}$ over variables in the signal set S and noise set S^c (including the intercept), respectively (which we call “average estimation error”); (ii) the average CI coverage probability over variables in the signal set S and noise set S^c; (iii) the average CI length over $j \in$ signal set S and noise set S^c. Note that there is no explicit formula of $\boldsymbol{\Theta}$ for logistic and Poisson regression models. Here we approximated it through 5 × 10^6 Monte-Carlo simulations. Notice that the average estimation error of $\mathscr{A}_h$-Trans-GLM declines as K increases, which agrees with our theoretical analysis in Section 3.3. As for the coverage probability, although CIs obtained by desparsified Lasso can achieve 95% coverage probability on S^c in linear and binomial cases, it fails to meet the 95% requirement of coverage probability on S in all three cases. In contrast, CIs provided by $\mathscr{A}_h$-Trans-GLM can achieve approximately 95% level when K is large on both S and S^c. Finally, the results of average CI length reveal that the CIs obtained by $\mathscr{A}_h$-Trans-GLM tend to be wider as K increases. Considering this together with the average estimation error and coverage probability, a possible explanation could be that desparsified Lasso might down-estimate $\boldsymbol{\Theta}$

$_{jj}$ which leads to too narrow CIs to cover the true coefficients. And $\mathscr{A}_h$-Trans-GLM offers a more accurate estimate of $\mathbf{\Theta}_{jj}$ which results in wider CIs.

We also consider different $(\{n_k\}_{k=0}^K, p, s)$ settings with the results in the supplements.

4.1.2 Transfer learning when $\mathscr{A}_h$ is unknown—

Different from the previous subsection, now we fix the total number of sources as $K = 10$. There are two types of sources, which belong to either $\mathscr{A}_h$ or $\mathscr{A}_h^c$. Sources from $\mathscr{A}_h$ have similar coefficients to the target one, while the coefficients of sources from $\mathscr{A}_h$ can be quite different. Intuitively, using more sources from $\mathscr{A}_h$ benefits the estimation of the target coefficient. But in practice, $\mathscr{A}_h$ may not be known as a priori. As we argued before, Trans-GLM can detect useful sources automatically, therefore it is expected to be helpful in such a scenario. Simulations in this section aim to justify the effectiveness of Trans-GLM.

Here is the detailed setting. We set the target sample size $n_0 = 200$ and source sample sample size $n_k = 200$ for all $k \neq 0$. The dimension $p = 2000$. Target coefficient is the same as the one used in Section 4.1.1 and we fix the signal number $s = 20$. Recall $\mathbf{R}_p^{(k)}$ denotes p independent Rademacher variables and $\mathbf{R}_p^{(k')}$ are independent for any $k \neq k2$. Consider $h = 20$ and 40. For any source data k in $\mathscr{A}_h$, we set $\mathbf{w}^{(k)} = \boldsymbol{\beta} + (h / p)\mathbf{R}_p^{(k)}$. For linear and logistic regression models, predictors from target $\mathbf{x}_i^{(0)} \overset{i.i.d.}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = [\Sigma_{jj2}]_{p \times p}$ where $\Sigma_{jj2} = 0.9^{|j-j2|}$, for all $i = 1, \ldots, n_0$. For the source, we generate p-dimensional predictors from independent t-distribution with degrees of freedom 4. For the target and sources of Poisson regression model, we generate predictors from the same Gaussian distribution and t-distribution respectively, and truncate each predictor at ±0.5.

To generate the coefficient $\mathbf{w}^{(k)}$ for $k \notin \mathscr{A}_h$, we randomly generate $S^{(k)}$ of size s from $\{2s + 1, \ldots, p\}$. Then, the j-th component of coefficient $\mathbf{w}^{(k)}$ is set to be

$$w_j^{(k)} = \begin{cases} 0.5 + 2hr_j^{(k)} / p, & j \in \{s+1, \ldots, 2s\} \cup S^{(k)}, \\ 2hr_j^{(k)} / p, & \text{otherwise,} \end{cases}$$

where $r_j^{(k)}$ is a Rademacher variable. We also add an intercept 0.5. The generating process of each source data is independent. Compared to the setting in Section 4.1.1, the current setting is more challenging because source predictors come from t-distribution with heavier tails than sub-Gaussian tails. However, although Assumption 2 is violated, in the following analysis, we will see that Trans-GLM can still succeed in detecting informative sources.

As before, we fit naïve-Lasso on only target data. $\mathscr{A}_h$-Trans-GLM and Pooled-Trans-GLM represent Algorithm 1 on source data in $\mathscr{A}_h$ and target data or all sources and target data, respectively. Trans-GLM runs Algorithm 2 by first identifying the informative source set $\mathscr{A}$, then applying Algorithm 1 to fit the model on sources in $\mathscr{A}$. We vary the values of $K_{\mathscr{A}_h}$ and h, and repeat simulations in each setting 200 times. The average ℓ_2-estimation errors are summarized in Figure 3.

From Figure 3, it can be observed that in all three models, $\mathscr{A}_h$-Trans-GLM always achieves the best performance as expected since it transfers information from sources in $\mathscr{A}_h$. Trans-GLM mimics the behavior of $\mathscr{A}_h$-Trans-GLM very well, implying that the transferable source detection algorithm can successfully recover $\mathscr{A}_h$. When $K_{\mathscr{A}_h}$ is small, Pooled-Trans-GLM performs worse than naïve-Lasso because of the negative transfer. As $K_{\mathscr{A}_h}$ increases, the performance of Pooled-Trans-GLM improves and finally matches those of $\mathscr{A}_h$-Trans-GLM and Trans-GLM when $K_{\mathscr{A}_h} = K = 10$.

4.2 A real-data study

In this section, we study the 2020 US presidential election results of each county. We only consider win or lose between two main parties, Democrats and Republicans, in each county. The 2020 county-level election result is available at https://github.com/tonmcg/US_County_Level_Election_Results_08-20. The response is the election result of each county. If Democrats win, we denote this county as class 1, otherwise, we denote it as class 0. And we also collect the county-level information as the predictors, including the population and race proportions, from https://www.kaggle.com/benhamner/2016-us-election.

The goal is to explore the relationship between states in the election using transfer learning. We are interested in swing states with a large number of counties. Among 49 states (Alaska and Washington, D.C. excluded), we select the states where the proportion of counties voting Democrats falls in [10%,90]%, and have at least 75 counties as target states. They include Arkansas (AR), Georgia (GA), Illinois (IL), Michigan (MI), Minnesota (MN), Mississippi (MS), North Carolina (NC), and Virginia (VA).

The original data includes 3111 counties and 52 county-level predictors. We also consider the pairwise interaction terms between predictors. After pre-processing, there are 3111 counties and 1081 predictors in the final data, belonging to 49 US states.

We would like to investigate which states have a closer relationship with these target states by our transferable source detection algorithm. For each target state, we use it as the target data and the remaining 48 states as source datasets. Each time we randomly sample 80% of target data as training data and the remaining 20% is used for testing. Then we run Trans-GLM (Algorithm 2) and see which states are in the estimated transferring set $\hat{\mathscr{A}}$. We repeat the simulation 500 times and count the transferring frequency between every state pair. The 25 (directed) state pairs with the highest transferring frequencies are visualized in Figure 4. Each orange node represents a target state we mentioned above and blue nodes are source states. States with the top 25 transferring frequencies are connected with a directed edge.

From Figure 4, we observe that Michigan has a strong relationship with other states, since there is a lot of information transferable when predicting the county-level results in Michigan, Minnesota, and North Carolina. Another interesting finding is that states which are geographically close to each other may share more similarities. For instance, see the connection between Indiana and Michigan, Ohio and Michigan, North Carolina and Virginia, South Carolina and Georgia, Alabama and Georgia, etc.

In addition, one can observe that states in the Rust Belt also share more similarities. As examples, see the edges among Pennsylvania, Minnesota, Illinois, Michigan, New York, and Ohio, etc.

To further verify the effectiveness of our GLM transfer learning framework on this dataset and make our findings more convincing, we calculate the average test misclassification error rates for each of the eight target states. For comparison, we compare the performances of Trans-GLM and Pooled-Trans-GLM with naïve-Lasso, SVM, trees, random forests (RF), boosting trees (Boosting) as well as their pooled version. Average test errors and the standard deviations of various methods are summarized in Table 1. The best method and other top three methods for each target are highlighted in bold and italics, respectively.

Table 1 shows that in four out of eight scenarios, Trans-GLM performs the best among all approaches. Moreover, Trans-GLM is always ranked in the top three except in the case of target state MS. This verifies the effectiveness of our GLM transfer learning algorithm. Besides, Pooled-Trans-GLM can always improve the performance of naïve-Lasso, while for other methods, pooling the data can sometimes lead to worse performance than that of the model fitted on only the target data. This marks the success of our two-step transfer learning framework and the importance of the debiasing step. Combining the results with Figure 4, it can be seen that the performance improvement of Trans-GLM (compared to naïve-Lasso) for the target states with more connections (share more similarities with other states) are larger. For example, Trans-GLM outperforms naïve-Lasso a lot on Michigan, Minnesota and North Carolina, while it performs similarly to naïve-Lasso on Mississippi.

We also try to identify significant variables by Algorithm 3. Due to the space limit, we put the results and analysis in Section S.1.3.4 of supplements. Interested readers can find the details there. Furthermore, since we have considered all main effects and their interactions, one reviewer pointed out that besides the classical Lasso penalty, there are other variants like group Lasso (Yuan and Lin, 2006) or Lasso with hierarchy restriction (Bien et al., 2013), which may bring better practical performance and model interpretation. To be consistent with our theories, we only consider the Lasso penalty here and leave other options for future study.

5 Discussions

In this work, we study the GLM transfer learning problem. To the best of our knowledge, this is the first paper to study high-dimensional GLM under a transfer learning framework, which can be seen as an extension to Bastani (2021) and Li et al. (2021). We propose GLM transfer learning algorithms, and derive bounds for ℓ_1/ℓ_2-estimation error and a prediction error measure with fast and slow rates under different conditions. In addition, to avoid the negative transfer, an algorithm-free transferable source detection algorithm is developed and its theoretical properties are presented in detail. Moreover, we accommodate the two-step transfer learning method to construct confidence intervals of each coefficient component with theoretical guarantees. Finally, we demonstrate the effectiveness of our algorithms via simulations and a real-data study.

There are several promising future avenues that are worth further research. The first interesting problem is how to extend the current framework and theoretical analysis to other models, such as multinomial regression and the Cox model. Second, Algorithm 1 is shown to achieve the minimax ℓ_1/ℓ_2 estimation error rate when the homogeneity assumption (Assumption 4) holds. Without homogeneity of predictors between target and source, only sub-optimal rates are obtained. This problem exists in the line of most existing high-dimensional transfer learning research (Bastani, 2021; Li et al., 2021, 2020). It remains unclear how to achieve the minimax rate when source predictors' distribution deviates a lot from the target one. Another promising direction is to explore similar frameworks for other machine learning models, including support vector machines, decision trees, and random forests.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank the editor, the AE, and anonymous reviewers for their insightful comments which have greatly improved the scope and quality of the paper.

This work was supported by NIH grant 1R21AG074205-01, NYU University Research Challenge Fund, and through the NYU IT High Performance Computing resources, services, and staff expertise.

References

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

Bien J, Taylor J, and Tibshirani R (2013). A lasso for hierarchical interactions. The Annals of Statistics, 41 (3):1111–1141. [PubMed: 26257447]

Cai TT and Wei H (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. The Annals of Statistics, 49(1):100–128.

Chen A, Owen AB, Shi M, et al. (2015). Data enriched linear regression. Electronic Journal of Statistics, 9(1):1078–1112.

Friedman J, Hastie T, and Tibshirani R (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1. [PubMed: 20808728]

Gross SM and Tibshirani R (2016). Data shared lasso: A novel tool to discover uplift. Computational Statistics & Data Analysis, 101:226–235. [PubMed: 29056802]

Hajiramezanali E and Zamani S (2018). Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. Advances in Neural Information Processing Systems 31 (NIPS 2018), 31.

Hanneke S and Kpotufe S (2020a). A no-free-lunch theorem for multitask learning. arXiv preprint arXiv:2006.15785.

Hanneke S and Kpotufe S (2020b). On the value of target data in transfer learning. arXiv preprint arXiv:2002.04747.

Kontorovich A (2014). Concentration in unbounded metric spaces and algorithmic stability. In International Conference on Machine Learning, pages 28–36. PMLR.

Li S, Cai TT, and Li H (2020). Transfer learning in large-scale gaussian graphical models with false discovery rate control. arXiv preprint arXiv:2010.11037.

Li S, Cai TT, and Li H (2021). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. Journal of the Royal Statistical Society: Series B (Statistical Methodology). To appear.

Loh P-L and Wainwright MJ (2015). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. The Journal of Machine Learning Research, 16(1):559–616.

McCullagh P and Nelder JA (1989). Generalized Linear Models, volume 37. CRC Press.

Meinshausen N and Bühlmann P (2006). High-dimensional graphs and variable selection with the lasso. The Annals of Statistics, 34(3):1436–1462.

Mendelson S, Pajor A, and Tomczak-Jaegermann N (2008). Uniform uncertainty principle for bernoulli and subgaussian ensembles. Constructive Approximation, 28(3):277–289.

Negahban S, Yu B, Wainwright MJ, and Ravikumar PK (2009). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In Advances in Neural Information Processing Systems, pages 1348–1356. Citeseer.

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

Pan SJ and Yang Q (2009). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359.

Plan Y and Vershynin R (2013). One-bit compressed sensing by linear programming. Communications on Pure and Applied Mathematics, 66(8):1275–1297.

Reeve HW, Cannings TI, and Samworth RJ (2021). Adaptive transfer learning. The Annals of Statistics, 49(6):3618–3649.

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

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

Van de Geer S, Bühlmann P, Ritov Y, and Dezeure R (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. The Annals of Statistics, 42(3):1166–1202.

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

Wang Z, Qin Z, Tang X, Ye J, and Zhu H (2018). Deep reinforcement learning with knowledge transfer for online rides order dispatching. In 2018 IEEE International Conference on Data Mining (ICDM), pages 617–626. IEEE.

Weiss K, Khoshgoftaar TM, and Wang D (2016). A survey of transfer learning. Journal of Big Data, 3(1):1–40.

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

Zheng C, Dasgupta S, Xie Y, Haris A, and Chen YQ (2019). On data enriched logistic regression. arXiv preprint arXiv:1911.06380.

Zhou S (2009). Restricted eigenvalue conditions on subgaussian random matrices. arXiv preprint arXiv:0912.4045.

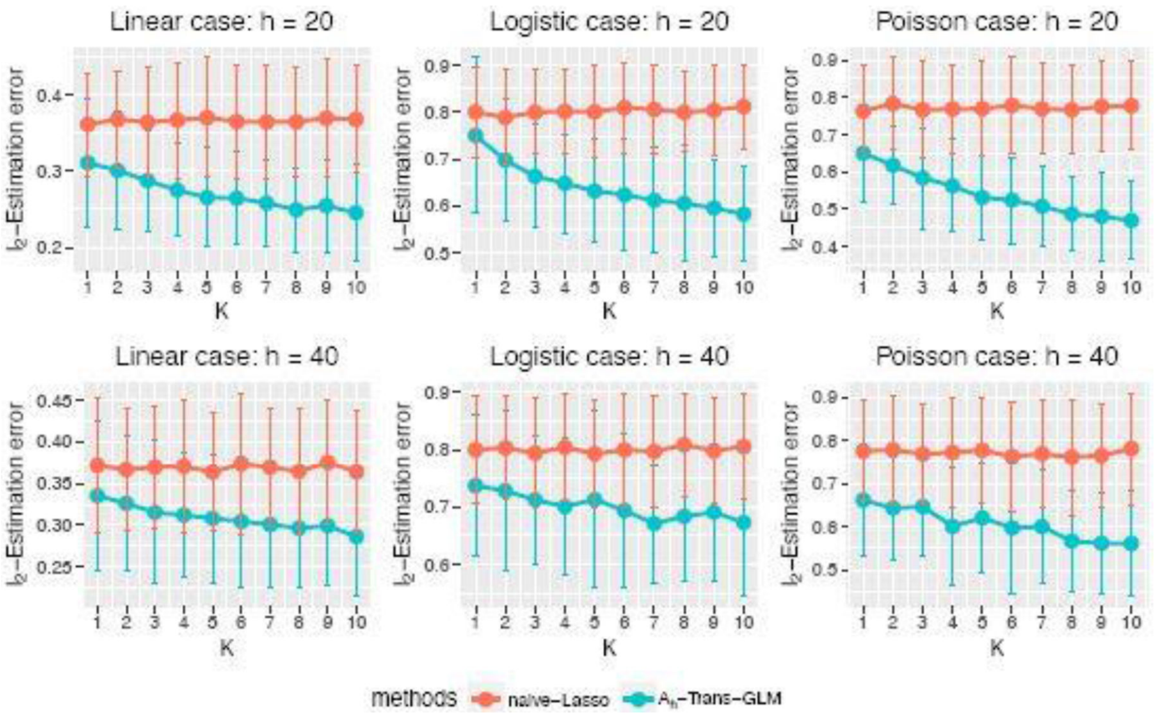

Fig. 1.

The average ℓ_2-estimation error of $\mathscr{A}_h$-Trans-GLM and naïve-Lasso under linear, logistic and Poisson regression models with different settings of h and K. $n_0 = 200$ and $n_k = 100$ for all $k = 1, \ldots, p$, $p = 500$, $s = 5$. Error bars denote the standard deviations.

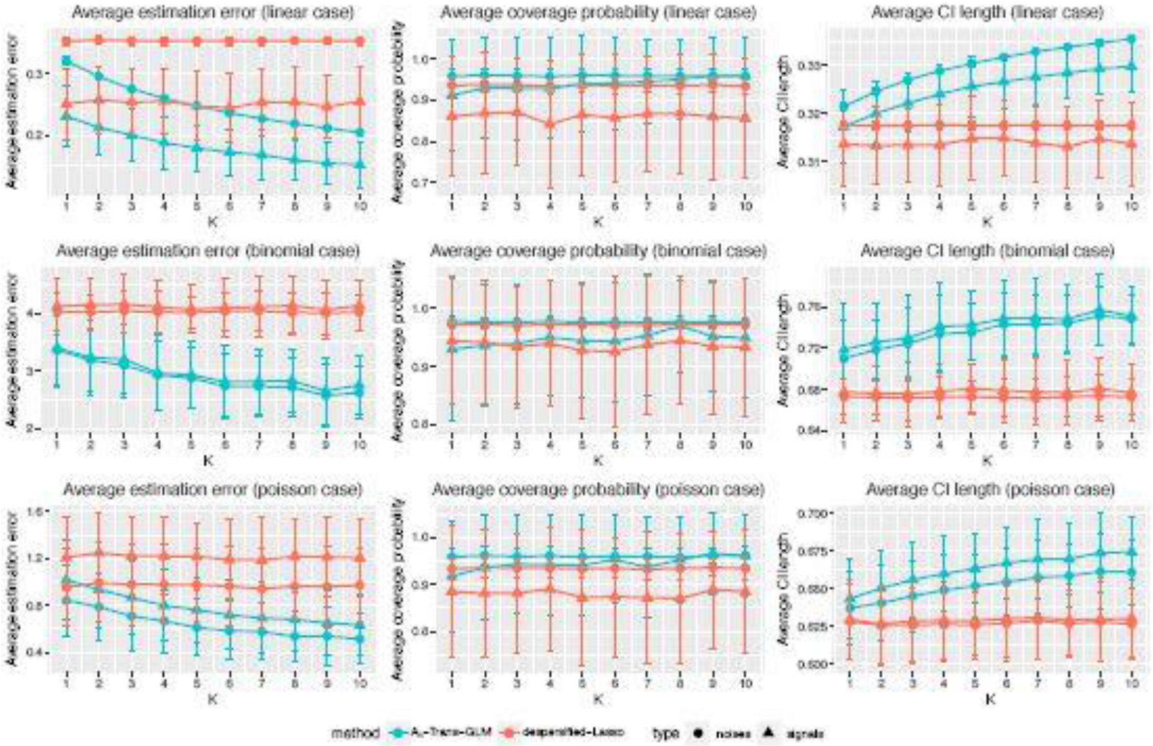

Fig. 2.

Three evaluation metrics of Algorithm 3 with $\mathscr{A}_h$ (we denote it as $\mathscr{A}_h$-Trans-GLM) and desparsified Lasso on target data, under linear, logistic and Poisson regression models, with different settings of K. $h = 20$. $n_0 = 200$ and $n_k = 100$ for all $k = 1, \ldots, p$, $p = 500$, $s = 5$. Error bars denote the standard deviations.

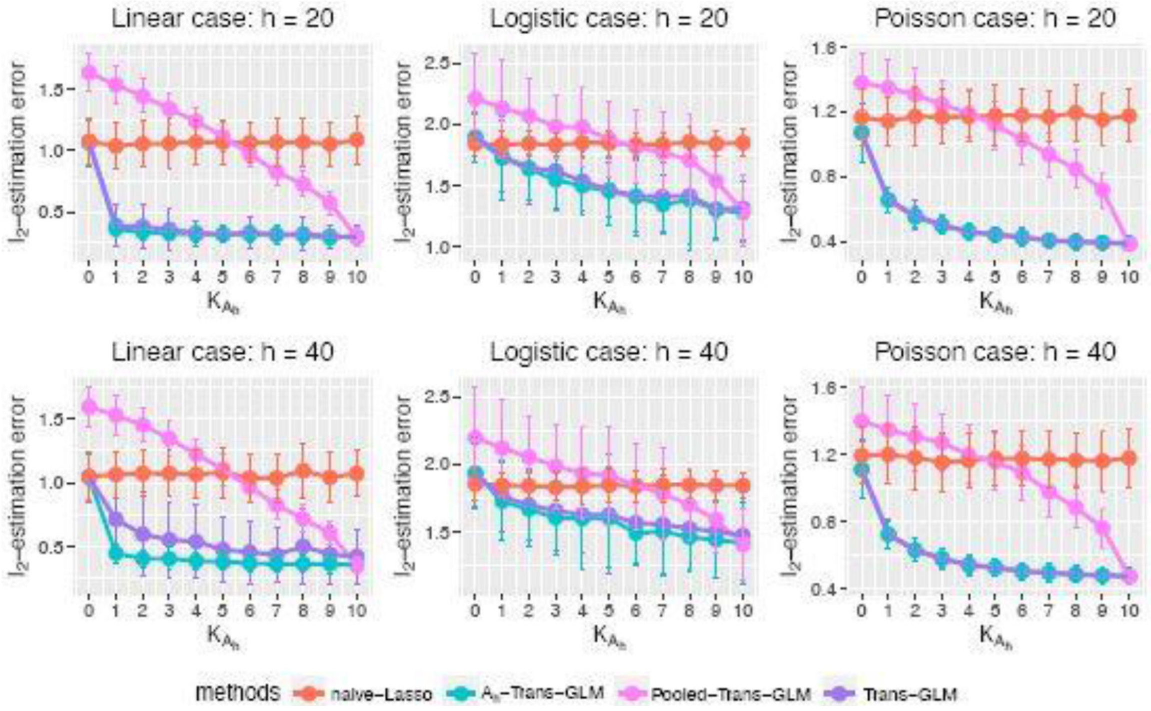

Fig. 3.

The average ℓ_2-estimation error of various models with different settings of h and $K_{\mathscr{A}_h}$ when $K = 10$. $n_k = 200$ for all $k = 0, \dots, K$, $p = 2000$, $s = 20$. Error bars denote the standard deviations.

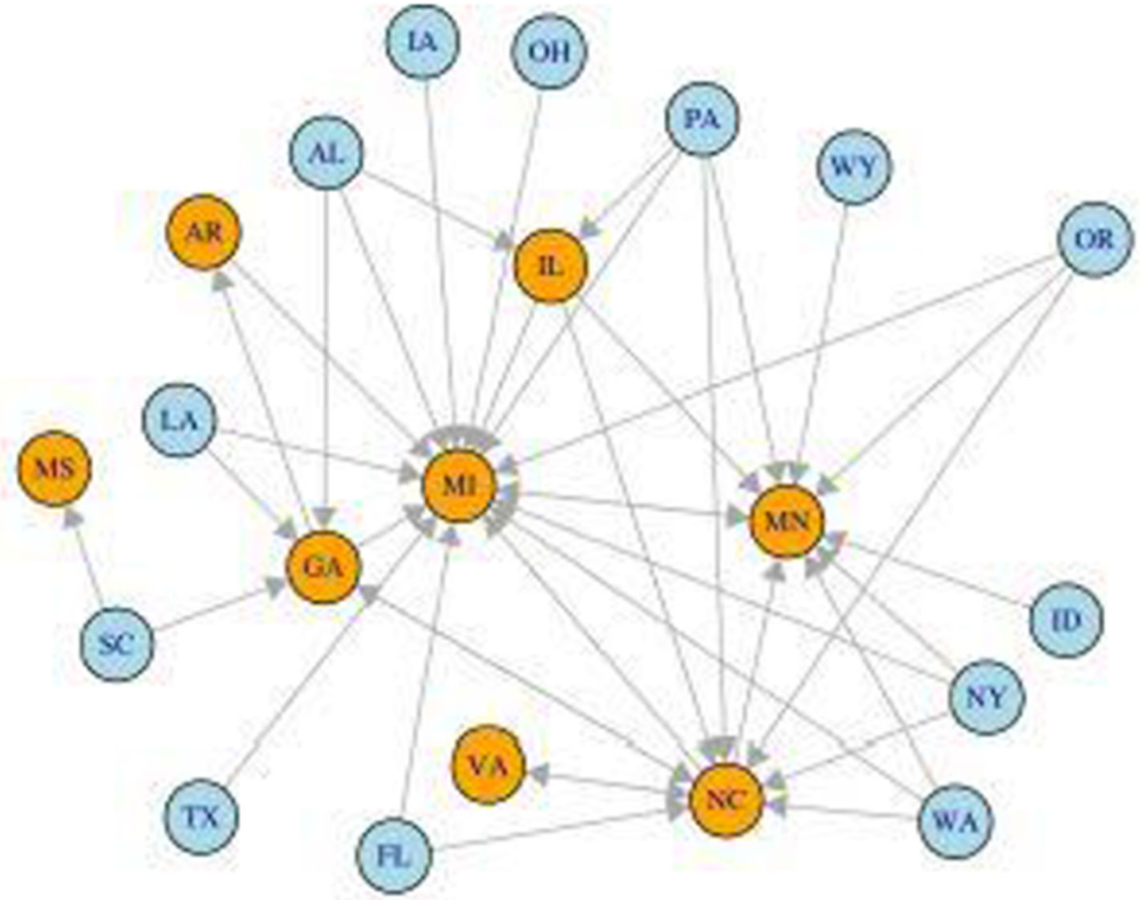

Fig. 4.

The transferability between different states for Trans-GLM.

Table 1

The average test error rate (in percentage) of various methods with different targets among 500 replications. The cutoff for all binary classification methods is set to be 1/2. Numbers in the subscript indicate the standard deviations.

Methods	**Target states**							
	AR	**GA**	**IL**	**MI**	**MN**	**MS**	**NC**	**VA**
naïve-Lasso	$4.79_{3.36}$	$6.98_{3.90}$	$5.73_{4.14}$	$11.49_{2.44}$	$12.46_{2.70}$	$7.53_{6.57}$	$15.60_{6.73}$	$9.48_{4.88}$
Pooled-Lasso	$3.59_{4.71}$	$9.98_{4.22}$	$7.89_{5.56}$	$7.04_{5.80}$	*10.38*$_{5.18}$	$22.01_{7.18}$	$12.73_{5.35}$	$21.44_{5.46}$
Pooled-Trans-GLM	*1.83*$_{3.12}$	*4.86*$_{3.60}$	*2.52*$_{3.55}$	*5.62*$_{4.54}$	$10.75_{5.60}$	*7.23*$_{6.65}$	*9.71*$_{5.75}$	**7.15**$_{4.23}$
Trans-GLM	**1.54**$_{2.94}$	*4.74*$_{3.54}$	**2.51**$_{3.45}$	*5.53*$_{4.73}$	**10.34**$_{5.73}$	$7.24_{6.81}$	**9.34**$_{5.57}$	*7.18*$_{4.67}$
SVM	$6.71_{1.70}$	$17.09_{3.89}$	$7.00_{5.40}$	$12.59_{1.87}$	$13.29_{2.29}$	$23.92_{8.90}$	$12.66_{6.86}$	$10.78_{5.29}$
Pooled-SVM	$7.84_{6.32}$	$13.47_{4.73}$	$7.75_{5.24}$	$7.58_{6.40}$	$13.01_{5.69}$	$27.32_{8.72}$	$12.30_{5.75}$	$17.31_{5.46}$
Tree	*2.23*$_{3.58}$	$8.37_{4.40}$	$4.62_{5.27}$	$10.05_{5.53}$	$10.97_{8.42}$	*5.97*$_{5.26}$	$18.29_{8.01}$	$14.46_{6.88}$
Pooled-Tree	$7.81_{6.89}$	$7.68_{4.59}$	$4.63_{4.26}$	$7.42_{6.18}$	$10.53_{5.91}$	$16.73_{7.30}$	$14.76_{7.26}$	$17.43_{5.85}$
RF	$3.60_{3.57}$	$6.04_{3.59}$	$4.08_{3.98}$	$6.42_{4.79}$	*10.51*$_{5.10}$	$7.27_{5.72}$	$11.29_{6.29}$	$7.73_{4.77}$
Pooled-RF	$3.73_{4.82}$	$7.49_{3.90}$	$4.35_{3.63}$	**5.34**$_{4.99}$	$10.86_{4.96}$	$12.56_{6.88}$	$11.04_{6.03}$	$10.40_{5.18}$
Boosting	$2.23_{3.58}$	**4.65**$_{3.77}$	*2.55*$_{3.82}$	$7.79_{5.52}$	$10.64_{6.51}$	**5.28**$_{5.16}$	$10.88_{6.47}$	*7.53*$_{5.10}$
Pooled-Boosting	$3.10_{4.84}$	$5.71_{3.53}$	$3.82_{3.85}$	$5.81_{5.27}$	$11.21_{5.13}$	$14.31_{7.42}$	*10.82*$_{5.99}$	$11.95_{5.25}$