Transfer Learning Under High-Dimensional Generalized Linear Models

Ye Tian & Yang Feng

To cite this article: Ye Tian & Yang Feng (2023) Transfer Learning Under High-Dimensional Generalized Linear Models, Journal of the American Statistical Association, 118:544, 2684-2697, DOI: 10.1080/01621459.2022.2071278

To link to this article: https://doi.org/10.1080/01621459.2022.2071278

Published online: 27 Jun 2022.

Article views: 5931

View related articles

View Crossmark data

Citing articles: 28 View citing articles
Transfer Learning Under High-Dimensional Generalized Linear Models

Ye Tian and Yang Feng

ABSTRACT

In this work, we study the transfer learning problem under high-dimensional 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 \( \ell_1/\ell_2 \) estimation error bounds as well as a bound for a prediction error measure. The theoretical analysis shows that when the target and sources 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. Supplementary materials for this article are available online.

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 additional data to help with the original target task motivates a well-known concept in computer science: transfer learning (Torrey and Shavlik 2010; Weiss, Khoshgoftaar, and Wang 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 article, 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 article, 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 \( \ell_0 \)-sparse. Li, Cai, and Li (2021) further explored the multi-source high-dimensional linear regression problem where both target and source samples are high-dimensional. The \( \ell_2 \)-estimation error bound under \( \ell_q \)-regularization (\( q \in [0,1] \)) was derived and proved to be minimax optimal under some conditions. In Li, Cai, and Li (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 nonparametric classification model (Cai and Wei 2021; Reeve, Cannings, and Samworth 2021) and the analysis under general

CONTACT Yang Feng yang.feng@nyu.edu Department of Biostatistics, School of Global Public Health, New York University, New York, NY

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.
functional classes via transfer exponents (Hanneke and Kpotufe 2020a, 2020b), etc. In addition, during the past few years, there have been some related works studying parameter sharing under the regression setting. For instance, Chen, Owen, and Shi (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 $\ell_1$ 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, Cai, and Li (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 $\ell_1$-sparse, which differs from the $\ell_0$-sparsity considered in Bastani (2021). The theoretical analysis shows that when the target and sources 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 initial 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 existing 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 article 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 $\ell_1$ and $\ell_2$-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 article. We use bold capitalized letters (e.g., $X$, $A$) to denote matrices, and use bold little letters (e.g., $x$, $y$) to denote vectors. For a $p$-dimensional vector $x = (x_1, \ldots, x_p)^T$, we denote its $\ell_q$-norm as $\|x\| = \left(\sum_{i=1}^{p}|x_i|^q\right)^{1/q}$ ($q \in (0, 2]$), and $\ell_0$-“norm” as $\|x\|_0 = \#\{j : x_j \neq 0\}$. For a matrix $A_{p \times q} = [a_{ij}]_{p \times q}$, its 1-norm, 2-norm, $\infty$-norm, and max-norm are defined as $\|A\|_1 = \sum_{j=1}^{q}|a_{ij}|$, $\|A\|_2 = \max_{1 \leq j \leq p} \|a_{ij}\|_2$, $\|A\|_\infty = \sup_{1 \leq j \leq q} \|a_{ij}\|_\infty$, and $\|A\|_{\max} = \sup_{1 \leq j \leq q} |a_{ij}|$, respectively. For two nonzero 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 $a_n = O(b_n)$ to represent $|a_n/b_n| \rightarrow 0$, where we defined the new command $"O"$ at the beginning to the text file as follows: when $a_n \ll b_n$ or $a_n = O(b_n)$ 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 $x_n \lesssim y_n$ or $x_n = O_p(y_n)$ means that for any $\epsilon > 0$, there exists a positive constant $M$ such that $\sup_n \mathbb{P}(|x_n/y_n| > M) \leq \epsilon$. And for two real numbers $a$ and $b$, we use $a \lor b$ and $a \land b$ to represent $\max(a, b)$ and $\min(a, b)$, respectively. Without specific notes, the expectation $\mathbb{E}$, variance $\text{var}$, and covariance $\text{cov}$ are calculated based on all randomness.

2.1. Generalized Linear Models (GLMs)

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

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

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

2.2. Target Data, Source Data, and Transferring Level

In this article, we consider the following multi-source transfer learning problem. Suppose we have the target dataset $(X^{(0)}, y^{(0)})$ and $K$ source datasets with the $k$th source denoted as $(X^{(k)}, y^{(k)})$, where $X^{(k)} \in \mathbb{R}^{n_k \times p}$, $y^{(k)} \in \mathbb{R}^{n_k}$ for $k = 0, \ldots, K$. The $i$th row of $X^{(k)}$ and the $i$th element of $y^{(k)}$ are denoted as $x^{(k)}_i$ and
\( y_j^{(k)} \), 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:

\[
y_j^{(k)} | x \sim P(y_j | x) = \rho(y_j) \exp\{x_j^T w^{(k)}_j - \psi(x_j^T w^{(k)}_j)\},
\]

for \( k = 0, \ldots, K \), with possibly different coefficient \( w^{(k)}_j \in \mathbb{R}^p \), the predictor \( x_j \in \mathbb{R}^p \), and some known univariate functions \( \rho \) and \( \psi \). Denote the target parameter as \( \beta = w^{(0)} \). Suppose the target model is \( \ell_0 \)-sparse, which satisfies \( \| \beta \|_0 = s \ll p \). This means that only \( s \) of the \( p \) variables contribute to the target response. Intuitively, if \( w^{(k)} \) is close to \( \beta \), the \( k \)th source could be useful for transfer learning.

Define the \( k \)th contrast \( \delta^{(k)} = \beta - w^{(k)} \). Given some \( A \subseteq \{1, \ldots, K\} \), we denote \( h = \max_{k \in A} \| \delta^{(k)} \| \) and we call this \( h \) as the transferring level of \( A \). We denote this index set as \( A_h \) to emphasize its associated transferring level. Note that in general, \( h \) can be any positive value. However, in our regime of interest, \( h \) shall be reasonably small to guarantee that transferring sources in \( A_h \) is beneficial. Denote \( n_{A_h} = \sum_{k \in A_h} n_k \), \( \alpha_k = \frac{n_k}{n_{A_h} + n_0} \) for \( k \in \{0\} \cup A_h \) and \( K_{A_h} = |A_h| \).

Note that in (1), we assume GLMs of the target and all sources share the same link function \( \psi \). 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 \( \psi \), 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 \( \psi \).

### 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 \( A \). The algorithm is motivated by the ideas in Bastani (2021) and Li, Cai, and Li (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 \( \ell_1 \)-penalty by pooled samples first, then fit the contrast in the second step using only the target by another \( \ell_1 \)-regularization. The detailed algorithm (\( A \)-Trans-GLM) is presented in Algorithm 1. The transferring step could be understood as to solve the following equation w.r.t. \( w \in \mathbb{R}^p \):

\[
\sum_{k \in \{0\} \cup A_h} \left[ (x_i^{(k)})^T (y_i^{(k)} - \sum_{i=1}^{n_k} \psi'((w_i)^T x_i^{(k)}) x_i^{(k)}) \right] = 0_p,
\]

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

\[
\sum_{k \in \{0\} \cup A_h} \alpha_k \mathbb{E} \left[ \psi'((w_i)^T x_i^{(k)}) - \psi'(w_i^0 x_i^{(k)}) | x_i^{(k)} \right] = 0_p,
\]

where \( \alpha_k = \frac{n_k}{n_{A_h} + n_0} \). Notice that in the linear case, \( w^A \) can be explicitly expressed as a linear transformation of the true parameter \( w^{(k)} \), that is, \( w^A = \Sigma^{-1} \sum_{k \in \{0\} \cup A_h} \alpha_k \Sigma^{(k)} w^{(k)} \), where

\[
\Sigma^{(k)} = \mathbb{E}[(x_i^{(k)})^T (x_i^{(k)})^T] \quad \text{and} \quad \Sigma = \sum_{k \in \{0\} \cup A_h} \alpha_k \mathbb{E}[x_i^{(k)} (x_i^{(k)})^T] \quad (Li, Cai, and Li 2021).
\]

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

#### Algorithm 1: \( A \)-Trans-GLM

**Input**: target data \( (X_0, y_0) \), source data \( (X^{(k)}, y^{(k)}) \), \( k = 1, \ldots, K \),\( \) penalty parameters \( \lambda_w \) and \( \lambda_\delta \), transferring set \( A \)

**Output**: the estimated coefficient vector \( \hat{\beta} \)

1. **Transferring step**: Compute \( \hat{w}^A \)
   
   \[
   \arg \min_w \left\{ \frac{1}{n_A + n_0} \sum_{k \in \{0\} \cup A} \left[ -(y_k)^T X^{(k)} w + \sum_{i=1}^{n_k} \psi(w_i^0 x_i^{(k)}) \right] + \lambda_w \| w \|_1 \right\}
   \]

2. **Debiasing step**: Compute \( \hat{\delta}^A \)
   
   \[
   \hat{\delta}^A \leftarrow \arg \min_\delta \left\{ -\frac{1}{n_0} (y_0)^T X_0 \hat{w}^A + \delta + \frac{1}{n_0} \sum_{i=1}^{n_0} \psi(w_i^0 + \delta x_i^{(0)}) + \lambda_\delta \| \delta \|_1 \right\}
   \]

3. Let \( \hat{\beta} \leftarrow \hat{w}^A + \hat{\delta}^A \)

4. Output \( \hat{\beta} \)

### 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, Khoshgoftaar, and Wang 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 \( \hat{A} \). 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, \( \{(X_0^{(r)}, y_0^{(r)})\}_{r=1}^3 \). 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 \( \hat{L}_0^{(k)} \) 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 \( \hat{L}_0^{(0)} \) is viewed as the loss of target. Finally, the difference between \( \hat{L}_0^{(k)} \) and \( \hat{L}_0^{(0)} \) is calculated and compared with some threshold, and sources with a difference less than the threshold will be recruited into \( \hat{A} \). 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 \( \mathbf{w} \), the average of negative log-likelihood on the \( r \)th fold of target data \((X^{(0)}[r], y^{(0)}[r])\) is

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

The detailed algorithm is presented as Algorithm 2.

**Algorithm 2: Trans-GLM**

1. **Input**: target data \((X^{(0)}, y^{(0)})\), all source data \\{(\((X^{(k)}), y^{(k)})_k\)\}, a constant \( C_0 > 0 \), penalty parameters \(\{\lambda^{(k)}[r]\}_{k=0}^{K-1} \)
2. **Output**: the estimated coefficient vector \(\hat{\mathbf{b}}\), and the determined transferring set \(\hat{\mathcal{A}}\)

1. **Transferable source detection**: Randomly divide \((X^{(0)}, y^{(0)})\) into three sets of equal size as \(\{(X^{(0)}[i], y^{(0)}[i])\}_{i=1}^{3}\)
2. for \(r = 1 \text{ to } 3\) do
3. \(\hat{\mathbf{b}}^{(0)}[r] \leftarrow \text{fit the Lasso on } \{(X^{(0)}[i], y^{(0)}[i])\}_{i=1}^{3}\) with penalty parameter \(\lambda^{(0)}[r]\)
4. \(\hat{\mathbf{b}}^{(k)}[r] \leftarrow \text{run Step 1 in Algorithm 1 with } \{(X^{(0)}[i], y^{(0)}[i])\}_{i=1}^{3} \cup (X^{(k)}, y^{(k)})\) and penalty parameter \(\lambda^{(k)}[r]\) for all \(k \neq 0\)
5. Calculate the loss function \(\hat{L}_0^{[r]}(\hat{\mathbf{b}}^{(k)}[r])\) on \((X^{(0)}[r], y^{(0)}[r])\) for \(k = 1, \ldots, K\)
6. end
7. \(\hat{\mathbf{b}}^{(k)} = \sum_{r=1}^{3} \hat{L}_0^{[r]}(\hat{\mathbf{b}}^{(k)}[r]) / 3, \hat{\mathbf{b}}^{(k)}_0 = \sum_{r=1}^{3} \hat{L}_0^{[r]}(\hat{\mathbf{b}}^{(0)}[r]) / 3, \hat{\sigma} = \sqrt{\sum_{r=1}^{3} \left(\hat{L}_0^{[r]}(\hat{\mathbf{b}}^{(0)}[r]) - \hat{L}_0^{[0]}(\hat{\mathbf{b}}^{(0)}[r])\right)^2 / 2, \hat{\mathcal{A}} \leftarrow \{k \neq 0 : \hat{L}_0^{[0]} - \hat{L}_0^{[k]} \leq C_0(\hat{\sigma} \vee 0.01)\}
8. **\(\text{A-Trans-GLM: } \hat{\mathbf{b}} \leftarrow \text{run Algorithm 1 using } \{(X^{(k)}, y^{(k)})\}_{k \in \hat{\mathcal{A}}} \)**
9. Output \(\hat{\mathbf{b}}\) and \(\hat{\mathcal{A}}\)

It’s important to point out that Algorithm 2 does not require the input of \(h\). We will show that \(\hat{\mathcal{A}} = \mathcal{A}_h\) for some specific \(h\) if certain conditions hold, in Section 3.2. Furthermore, under these conditions, transferring with \(\hat{\mathcal{A}}\) 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 \(\hat{\mathbf{b}}\) from the two-step transfer learning approach. In this section, we would like to construct the asymptotic confidence interval (CI) for each component of \(\hat{\mathbf{b}}\) 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 \(\hat{\mathbf{b}}\) 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 \(\mathbf{W} \in \mathbb{R}^{p^2}\), denote \(W_{w} = \text{diag}(\psi''(x_1^{(k)})^{T}w), \ldots, \psi''(x_{p_0}^{(k)})^{T}w))\), \(X_{w} = W_{w}X^{(k)}, \Sigma_{w} = \mathbb{E}[x^{(k)}(x^{(k)})^{T}\psi''(x^{(k)})^{T}w)]\) and \(\Sigma_{w}^{-1} = n_{k}^{-1}(X_{w}X_{w})^{T}X_{w}. X_{w}^{(k)}\) represents the \(j\)th column of \(X_{w}^{(k)}\) and \(X_{w,j}^{(k)}\) represents the matrix \(X_{w}^{(k)}\) without the \(j\)th column. \(\hat{\Sigma}_{w,j}^{(k)}\) represents the \(j\)th row of \(\hat{\Sigma}_{w}^{(k)}\) without the diagonal (\(j,j\)) element, and \(\hat{\Sigma}_{w,j}^{(k)}\) is the diagonal (\(j,j\)) element of \(\hat{\Sigma}_{w}^{(k)}\).

Next, we explain the details of the CI construction procedure in Algorithm 3. In Step 1, we obtain a point estimator \(\hat{\mathbf{b}}\) from \(\hat{\mathcal{A}}\)-Trans-GLM (Algorithm 1), given a specific transferring set \(\hat{\mathcal{A}}\). Then in Steps 2–4, we estimate the target inverse Fisher information matrix \(\Sigma^{-1}_{b} = \Sigma^{-1}_{b} = \Sigma^{-1}_{b}\) as

\[
\hat{\Sigma}_{b} = \text{diag}(\hat{\sigma}^{-2}_1, \ldots, \hat{\sigma}^{-2}_p) = \begin{pmatrix}
1 & -\hat{\gamma}_{1,2} & \cdots & -\hat{\gamma}_{1,p} \\
-\hat{\gamma}_{2,1} & 1 & \cdots & -\hat{\gamma}_{2,p} \\
\vdots & \vdots & \ddots & \vdots \\
-\hat{\gamma}_{p,1} & -\hat{\gamma}_{p,2} & \cdots & 1
\end{pmatrix},
\]

Finally in Step 5, we “debias” \(\hat{\mathbf{b}}\) using the target data to get a new point estimator \(\hat{b}\) which is asymptotically unbiased as

\[
\hat{b} = \hat{\mathbf{b}} + \frac{1}{n_0} \hat{\Theta}(X^{(0)})^{T} [Y^{(0)} - \psi'(X^{(0)}\hat{\mathbf{b}})],
\]

where \(\psi'(X^{(0)}\hat{\mathbf{b}}) := (\psi'(x_1^{(0)})^{T}\hat{\mathbf{b}}, \ldots, \psi'(x_{p_0}^{(0)})^{T}\hat{\mathbf{b}})^{T} \in \mathbb{R}^{p_0}\).

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, for example, random design in different replications (which we focus on in this article), multiple hypothesis testing might be more challenging.

### 3. Theory

In this section, we will establish theoretical guarantees on the three proposed algorithms. Section 3.1 provides a detailed anal-
Algorithm 3: Confidence interval construction via node-wise regression

Input: target data \((X^{(0)}, y^{(0)})\), source data \((X^{(k)}, y^{(k)})\) for \(k = 1, \ldots, K\), penalty parameters \((\lambda_j)^{(p)}_{p=1}\) and \((\tilde{\lambda}_j)^{(p)}_{p=1}\), transferring set \(\mathcal{A}\), confidence level \((1 - \alpha)\).

Output: Level \((1 - \alpha)\) confidence interval \(I_j\) for \(\beta_j\) with \(j = 1, \ldots, p\).

1. Compute \(\hat{\beta}\) via Algorithm 1.
2. Compute \(\hat{y}_j^{(A)}\) via Algorithm 1.
3. Compute \(\hat{\gamma}_j\) via Algorithm 1.
4. Compute \(\hat{\Sigma}_{j} = \hat{\gamma}_j\).
5. Compute \(I_j\) for \(j = 1, \ldots, p\).
6. Output \(I_j\) for \(j = 1, \ldots, p\).

Analysis of Algorithm 1 with transferring set \(\mathcal{A}\), which we denote as \(\mathcal{A}\)-Trans-GLM. Section 3.2 introduces certain conditions, under which we show that the transferring set \(\hat{\mathcal{A}}\) detected by Algorithm 2 (Trans-GLM) is equal to \(\mathcal{A}\) for some \(h\) with high probability. Section 3.3 presents the analysis of Algorithm 3 with transferring set \(\mathcal{A}\), where we prove a central limit theorem. For the proofs and some additional theoretical results, refer to supplementary materials.

3.1 Theory on \(\mathcal{A}\)-Trans-GLM

We first impose some common assumptions about GLM.

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

Assumption 2. For any \(a \in \mathbb{R}^p\), \(a^T x^{(k)}\)'s are iid \(\kappa_u a^T a\)-sub-Gaussian variables with zero mean for all \(k = 0, \ldots, K\), where \(\kappa_u\) is a positive constant. Denote the covariance matrix of \(x^{(k)}\) as \(\Sigma^{(k)}\), with \(\inf_k \lambda_{\text{min}}(\Sigma^{(k)}) \geq \kappa_t > 0\), where \(\kappa_t\) is a constant.

Assumption 3. At least one of the following assumptions hold: \((M_\phi, U)\) are some positive constants)

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

Assumption 4. Denote \(\hat{\Sigma}^{(k)} = \sum_{k=0}^{K} \alpha_k E\left[\int_0^1 \psi''((x^{(k)})^T \beta + t(x^{(k)})^T (w^{(k)} - \beta)) dt \cdot x^{(k)}(x^{(k)})^T\right]\) and \(\hat{\Sigma}_{\infty}^{(k)} = E\left[\int_0^1 \psi''((x^{(k)})^T \beta + t(x^{(k)})^T (w^{(k)} - \beta)) dt \cdot x^{(k)}(x^{(k)})^T\right]\). It holds that sup_{k=0}^{K} \|\hat{\Sigma}^{(k)}\|_1 < \infty.

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

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, Cai, and Li (2021). Now, we are ready to present the following main result for the \(\mathcal{A}\)-Trans-GLM algorithm. Given some \(\mathcal{A} \subseteq \{1, \ldots, K\}\), define the parameter space as

\[\mathcal{E}(s, h) = \left\{ (\beta, (W^{(k)})_{k=1}^{K} : \|\beta\|_0 \leq s, \sup_{k \in \mathcal{A}} \|W^{(k)} - \beta\|_1 \leq h \right\}\]

Again, we denote the index set \(\mathcal{A}\) as \(\mathcal{A}_h\) to emphasize its associated transferring level.

Theorem 1 (\(\ell_1/\ell_2\)-estimation error bound of \(\mathcal{A}\)-Trans-GLM with Assumption 4). Assume Assumptions 1, 2, 3, and 4 hold.

Suppose \(h \leq \sqrt{\frac{n_0 \log p}{n_0 \log(1 + p)}}\) holds or Assumption 3. (ii) with \(h \leq C'U^{-1}U\) for some \(C' > 0\) holds. We take \(\lambda_w = C_w \frac{\log p}{n_0 \log(1 + p)}\) and \(\lambda_s = C_s \frac{\log p}{n_0 \log(1 + p)}\), where \(C_w\) and \(C_s\) are sufficiently large positive constants. Then
Theorem 2 indicates that under conditions on Remark 3.

Remark 2. When $h \ll \sqrt{\log p/n_0}$, the upper bounds in (6) and (7) are better than the classical Lasso $\ell_2$-bound $O_p\left(\frac{\log p}{n_0}\right)$ and $\ell_1$-bound $O_p\left(\frac{\log p}{n_0}\right)$ using only target data.

Similar to Theorem 2 in Li, Cai, and Li (2021), we can show the following lower bound of $\ell_1/\ell_2$-estimation error in regime $\Xi(s,h)$ in the minimax sense.

**Theorem 2 ($\ell_1/\ell_2$-minimax estimation error bound).** Assume Assumptions 1, 2, and 4 hold. Also assume Assumption 3.(i) holds or Assumption 3.(ii) with $n_0 \gtrsim s^2 \log p$ holds. Then
\[
\inf_{\hat{\beta}} \sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\|\hat{\beta} - \beta\|_2 \gtrsim \left(\frac{s \log p}{n_0} + n_0\right)^{1/2} + \left(\frac{\log p}{n_0} \right)^{1/4} h^{1/2} \right) \geq \frac{1}{2},
\]
\[
\inf_{\hat{\beta}} \sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\|\hat{\beta} - \beta\|_1 \gtrsim s \left(\frac{\log p}{n_{A_h} + n_0}\right)^{1/2} + h \right) \geq \frac{1}{2}.
\]

**Remark 3.** Theorem 2 indicates that under conditions on $h$ required by Theorem 1 ($h \lesssim s \sqrt{\log p/n_0}$), $A_h$-Trans-GLM achieves the minimax optimal rate of $\ell_1/\ell_2$-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 ($\ell_1/\ell_2$-estimation error bound of $A_h$-Trans-GLM without Assumption 4).** Assume Assumptions 1 and 2 hold. Suppose $h \ll \sqrt{\frac{n_0}{\log p}}$, $h \lesssim C^{-1/2} n_0 \gtrsim C\log p$ and $n_{A_h} \gtrsim C\log p$, 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 $\delta_w = C_w \left(\frac{\log p}{n_{A_h} + n_0} + h\right)$ and $\delta_\lambda = C_\lambda \sqrt{\frac{\log p}{n_0}}$, where $C_w$ and $C_\lambda$ are sufficiently large positive constants. Then
\[
\sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\|\hat{\beta} - \beta\|_2 \lesssim \left(\frac{s \log p}{n_{A_h} + n_0}\right)^{1/2} + \sqrt{sh} \right) \leq \left(\frac{\log p}{n_0} \right)^{1/4} h^{1/2} \wedge h \geq 1 - n_0^{-1},
\]
\[
\sup_{\xi \in \Xi(s,h)} \mathbb{P}\left(\|\hat{\beta} - \beta\|_1 \lesssim s \left(\frac{\log p}{n_{A_h} + n_0}\right)^{1/2} + h \right) \geq 1 - n_0^{-1}.
\]

**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 $O_p\left(\frac{\log p}{n_0}\right)$ with target data.

Comparing the results in Theorems 1 and 3, we know that with Assumption 4, we could get sharper $\ell_1/\ell_2$-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 $A_h$ for some specific $h$ that is, $\hat{A} = A_h$ with high probability. Under these conditions, transferring with $\hat{A}$ 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 $L_0^r[\hat{r}](w)$ defined in (3) is
\[
L_0(w) = -E[\log \rho(y^{(0)})] - E[y^{(0)}w^T x^{(0)}] + E[\psi(w^T x^{(0)})] = 0,
\]
\[
\text{where } \alpha_0(k) = \frac{2m}{2n_0/3 + n_k} \text{ and } \alpha_k = \frac{n_0}{2n_0/3 + n_k}.
\]

Define $\beta(0) = \beta$. Next, let's impose a general assumption to ensure the identifiability of some $A_h$ by Trans-GLM.

**Assumption 5 (Identifiability of $A_h$).** Denote $A_h^c = \{1, \ldots, K\} \setminus A_h$. Suppose for some $h$ and for all $k$, we have
\[
\mathbb{P}\left(\sup_r \mathbb{E}[\hat{L}_0^r](\hat{\beta}(k)|r) - \gamma_1^r(\beta(k)) > \gamma_2^r(\hat{\beta}(k)|r)\right) \lesssim \hat{g}_1^r(\xi),
\]
\[
\mathbb{P}\left(\sup_r \mathbb{E}[\hat{L}_0^r](\beta(k)|r) - \gamma_1^r(\beta(k)|r) > \gamma_2^r(\beta(k)|r)\right) \lesssim \hat{g}_2^r(\xi),
\]
where $\hat{g}_1^r(\xi), \hat{g}_2^r(\xi) \to 0$ as $\xi \to \infty$. Assume $\inf_{k \in A_h^c} \frac{\delta}{\xi}(\xi) = \tilde{\lambda} > 0$, and

\[
\psi''((1 - t)(x^{(0)} + \beta^T x^{(0)})^T \beta - t(x^{(0)} + \beta^T x^{(0)})^T \beta^T dt \cdot x^{(0)}(x^{(0)} + \beta^T x^{(0)})^T) := \hat{\lambda} > 0,
\]

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

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

where \( C_1 > 0 \) is sufficiently large.

**Remark 5.** Here we use generic notations \( \Upsilon_1^{(k)} \), \( \Gamma_1^{(k)} \), \( \Gamma_2^{(k)} \), \( \delta_1^{(k)}(\xi) \), and \( \delta_2^{(k)}(\xi) \). We show their explicit forms under linear, logistic, and Poisson regression models in Proposition 1 in Section S.1.2.1 of supplementary materials.

**Remark 6.** Condition (8) guarantees that for the sources not in \( \mathcal{A}_h \), 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 \( \sup_{r} L_0^{(r)}(\beta^{(k)}) - L_0^{(r)}(\beta^{(k)}) \) and \( \sup_{r} L_0^{(r)}(\beta^{(k)}) - L_0^{(r)}(\beta^{(k)}) \) 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 \( \mathcal{A}_h \)).** For Algorithm 2 (Trans-GLM), with Assumption 5 satisfied for some \( h \), for any \( \delta > 0 \), there exist constants \( C'(\delta) \) and \( N(\delta) > 0 \) such that when \( C_0 = C'(\delta) \) and \( \min_{k \in \{0,1\}; \mathcal{A}_h} n_k > N(\delta) \),
\[
\mathbb{P}(\hat{\mathcal{A}} = \mathcal{A}_h) \geq 1 - \delta.
\]

Then Algorithm 2 has the same high-probability upper bounds of \( \ell_1/\ell_2 \)-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 \( \hat{\mathcal{A}} \) suggested by Trans-GLM will be \( \mathcal{A}_h \) 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 sparsity 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 \( w^{(k)} \) and \( \beta^{(k)} \) with \( k \in \mathcal{A}_h^c \) for some \( h \). Note that the source data in \( \mathcal{A}_h \) automatically satisfy the sparsity condition due to the definition of \( \mathcal{A}_h \).

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

(i) \( w^{(k)} = \zeta^{(k)} + \vartheta^{(k)} \), where \( \| \zeta^{(k)} \|_0 \leq s' \) and \( \| \vartheta^{(k)} \|_1 \leq h' \);
(ii) \( \beta^{(k)} = \iota^{(k)} + \omega^{(k)} \), where \( \| \iota^{(k)} \|_0 \leq s' \) and \( \| \omega^{(k)} \|_1 \leq h' \).

**Corollary 1.** Assume Assumptions 1, 2, 6, and \( \inf_{k \in \mathcal{A}_h^c} \lambda^\min_{\mathbb{E}[\int_0^1 \psi''((1-t)(x^{(0)})^T \beta + t(x^{(0)})^T \beta^{(k)}) dt \cdot x^{(0)}(x^{(0)})^T] := \lambda > 0 \) hold. Also assume \( \sup_{k \in \mathcal{A}_h^c} \| \beta^{(k)} \|_\infty < \infty \) and \( \sup_k \| w^{(k)} \|_\infty < \infty \).
Let \( \lambda^{(k)}[r] = C \left( \sqrt{\frac{\log p}{n_k + n_0}} + h \right) \) when \( k \in \mathcal{A}_h \), \( \lambda^{(k)}[r] = C \left( \sqrt{\frac{\log p}{n_k + n_0}} \right) \) when \( k \in \mathcal{A}_h^c \) and \( \lambda^{(0)}[r] = C \left( \sqrt{\frac{\log p}{n_k + n_0}} \right) \) 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 \( \Omega = \sqrt{h} \left[ \left( \frac{\log p}{n_k + n_0} \right)^{1/4} + \left( \frac{\log p}{\min_{k \in \mathcal{A}_h^c} (n_k + n_0)} \right)^{1/4} \left( s' \vee s \right) \left( \frac{1}{1/4} + \sqrt{\frac{h}{(s' \vee s) \vee h}} \right) \right] \).

(i) For logistic regression models, we require
\[
\inf_{k \in \mathcal{A}_h} \| \beta^{(k)} - \beta \|_2 \geq \left( \frac{2 \sqrt{\log p}}{n_0} \right)^{1/4} \vee 1 + \Omega, \quad h \ll s^{-1/2}.
\]

(ii) For linear models, we require
\[
\inf_{k \in \mathcal{A}_h} \| \beta^{(k)} - \beta \|_2 \geq \left( \frac{s \vee h} {n_0} \right)^{1/4} \vee 1 + \Omega, \quad h \ll s^{-1}.
\]

(iii) For Poisson regression models, we require
\[
\inf_{k \in \mathcal{A}_h} \| \beta^{(k)} - \beta \|_2 \geq \left( \frac{s \vee h} {n_0} \right)^{1/4} \vee 1 + \Omega, \quad h \ll s^{-1}.
\]

Under Assumptions 1, 2, and the sufficient conditions derived in Corollary 1, by Theorem 4, we can conclude that \( \hat{\mathcal{A}} = \mathcal{A}_h \) for some \( h \). Note that we don’t impose Assumption 4 here. Remark 4 indicates that, for \( \mathcal{A}_h \)-Trans-GLM to have a faster convergence rate than Lasso on target data, we need \( h \ll \frac{\log p}{n_0} \) and \( n_{\mathcal{A}_h} \gg n_0 \). Suppose \( s' \approx s \) and \( h' \ll s'^{1/2} \). Then for logistic regression models, when \( n_0 \gg s^2 \log(p \vee K) \), we can claim that as long as the source is “helpful,” i.e., there exists such an \( h \ll \frac{\log p}{n_0} \), Trans-GLM is able to detect \( \mathcal{A}_h \). The same conclusion holds for linear models when \( n_0 \gg s^2 \log(p \vee K) \) and for Poisson models when \( n_0 \gg (s \log K) \vee (s^2 \log p) \). This implies that when target sample size \( n_0 \) is large enough and there exists an \( h \) such that \( \mathcal{A}_h \) is helpful, Trans-GLM can successfully
detect it with high probability and subsequently lead to a better \( \ell_1 / \ell_2 \)-estimation error bound than that of 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 \( \Sigma^{(k)}_\beta = \mathbb{E}[x^{(k)}(x^{(k)})^T \psi''((x^{(k)})^T \beta)] \). Let \( \Theta = (\Sigma^{(0)}_\beta)^{-1} \) and \( K_{A_h} = |A_h| \). Define

\[
\gamma_j^{(k)} = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E} \left\{ \psi''(\beta^T x^{(k)}) \cdot (x_j^{(k)} - (x_{j-1}^{(k)})^T \gamma_j^{(k)})^2 \right\}
\]

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

\[
\sup_{k \in A_h, j=1:p} \| (\Sigma^{(k)}_{\beta,j-j} - \Sigma^{(0)}_{\beta,j-j})^{-1} (\Sigma^{(0)}_{\beta,j-j} - \Sigma^{(0)}_{\beta,j-j}) \gamma_j^{(k)} - \gamma_j^{(0)} \|_1 \leq h_1,
\]

Then by the definition of \( \gamma_j^{(k)} \),

\[
\sup_{k \in A_h, j=1:p} \| \gamma_j^{(k)} - \gamma_j^{(0)} \|_1 \leq h_1,
\]

which is similar to our previous setting \( \sup_{x \in \mathbb{R}^p} \| w - \beta \|_1 \leq h \). This motivates us to apply a similar two-step transfer learning procedure (Steps 2–4 in Algorithm 3) to learn \( \gamma_j^{(0)} \) for \( j = 1, \ldots, p \). We impose the following set of conditions.

**Assumption 7.** Suppose the following conditions hold:

(i) \( \sup_{k \in \mathcal{O} \cup A_h} \| x^{(k)} \| \leq U < \infty, \sup_{k \in \mathcal{O} \cup A_h} \| (x^{(k)})^T w^{(k)} \| \leq U' < \infty \) a.s.;

(ii) \( \sup_{p \in \mathcal{P}, k \in \mathcal{O} \cup A_h} \| (x^{(k)})^T \gamma_j^{(k)} \| \leq U'' < \infty \) a.s.;

(iii) \( \inf_{k \in \mathcal{O} \cup A_h} \gamma_{\min} (\Sigma^{(k)}_{\beta,j-j}) \geq U > 0 \);

(iv) \( \sup_{k \in \mathcal{O} \cup A_h} \sup_{z \in \mathcal{U}} \psi''(x^{(k)})^T w^{(k)} + z \leq M_{\psi} < \infty \) a.s.

(v) \( \sup_{k \in \mathcal{O} \cup A_h} \| (\Sigma^{(k)}_{\beta,j-j})^{-1} \Sigma^{(0)}_{\beta,j-j} \|_1 \leq \infty \), where \( \Sigma_\beta = \sum_{k \in \mathcal{O} \cup A_h} \alpha_k \Sigma^{(k)}_\beta \);

(vi) \( \min_{k \in A_h} \alpha_k \gg n_0, n_0 \gg s^2 \log p, \)

(vii) \( h_1 \leq s^{-1/2} \left[ \frac{n_0}{\log p} \left( \frac{K_{A_h}}{s} \right) \right] \), \( h_1 \max \leq \frac{n_0^{1/4}}{s^{(\log p)^{1/4}}} \), \( h \leq \frac{n_0^{1/4}}{(s^2 \log p)^{1/4}} \), \( h_1 \max \leq s^{-1/2} \left( \frac{n_0^{1/4}}{s^{(\log p)^{1/4}}} \right) \), \( h_1 \ll \frac{n_0^{-3/4}}{(s^2 \log p)^{1/4}} \), \( h \ll \frac{n_0^{-1/4}}{(s^2 \log p)^{1/4}} \).

**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 \( s_j = \| \gamma_j^{(0)} \|_0 \) 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 \( \gamma_j^{(0)} \) 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 \( h, h_1, h_\max \), and the rate (11) may be worse. We do not explore the details in this article 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), \( \min_{k \in A_h} \alpha_k \gg n_0 \) is not necessary and the only reason we add it here is to simplify condition (vii).

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

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

\[
\frac{\sqrt{n_0}(\beta - \hat{\beta})}{\sqrt{\Theta_0^T \Sigma_\beta \Theta_0}} \sim \mathcal{N}(0,1),
\]

and

\[
\sqrt{\frac{\log p}{n_{A_h} + n_0}} + \sqrt{s} \left( h_1^{1/2} \left( \frac{\log p}{n_0} \right)^{1/4} \right) \right) \right) \right) + (s h_1)^{1/2} \left( \left( \frac{s \log p}{n_{A_h} + n_0} \right)^{1/4} \left( h_1^{1/4} \left( \frac{\log p}{n_0} \right)^{1/8} \right) \left( h_1^{1/2} \right) \right)
\]

for \( j = 1, \ldots, p \), with probability at least \( 1 - K_{A_h} n_0^{-1} \).
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 (11) 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 $h \ll \sqrt{\log p/n_0}, h_1 \ll \sqrt{s \log p/n_0}$. \[ \left[ s^{1/2} - \left( \frac{\kappa_0 + s}{n_0} \right)^{1/4} \right], s^{1/2} h^{1/4} \ll s^{1/2} \left( \frac{\log p}{n_0} \right)^{3/8} \text{ and } h_{\max} \ll \sqrt{\frac{\log p}{n_0}}, \] 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 data. In the simulation part, we study the performance of the Trans-GLM methods include Trans-GLM and Trans-GLM (Algorithm 2), naive-Lasso (Lasso on target data), $A_h$-Trans-GLM (Algorithm 1 with $A = A_h$) and Pooled-Trans-GLM (Algorithm 1 with all sources). In the real-data study, besides naive-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 supplementary materials.

4.1. Simulations

4.1.1. Transfer Learning on $A_h$

In this section, we study the performance of $A_h$-Trans-GLM and compare it with that of naive-Lasso. The purpose of the simulation is to verify that $A_h$-Trans-GLM can outperform naive-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 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 $\beta = (0.5 \cdot \mathbf{1}_s, 0_{p-s})^T$, where $\mathbf{1}_s$ has all $s$ elements 1 and $0_{p-s}$ has all ($p-s$) elements 0, where $s$ is set to be 5. Denote $\mathcal{R}_p^{(k)}$ as $p$ independent Rademacher variables (being $-1$ or 1 with equal probability) for any $k$. $\mathcal{R}_p^{(k)}$ is independent with $\mathcal{R}_p^{(k′)}$ for any $k \neq k′$. For any source data $k$ in $A_h$, we set $w^{(k)} = \beta + (h/p)\mathcal{R}_p^{(k)}$. For linear and logistic regression models, predictors from target $x_i^{(0)} \sim \mathcal{N}(0_p, \Sigma)$ with $\Sigma = [\Sigma_{ij}]_{p \times p}$ where $\Sigma_{ij} = 0.5^{||i-j||}$, for all $i = 1, \ldots, n$. And for $k \in A_h$, we generate $p$-dimensional predictors from $\mathcal{N}(0_p, \Sigma + \varepsilon\varepsilon^T)$, where $\varepsilon \sim \mathcal{N}(0_p, 0.3^2I_p)$ 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 $\pm 0.5$.

Note that naive-Lasso is fitted on only target data, and $A_h$-Trans-GLM denotes Algorithm 1 on source data in $A_h$ as well as target data. We train naive-Lasso and $A_k$-Trans-GLM models under different settings of $h$ and $K_{A_h}$, then calculate the $\ell_2$-estimation error of $\beta$. All the experiments are replicated 200 times and the average $\ell_2$-estimation error of $A_h$-Trans-GLM and naive-Lasso under linear, logistic, and Poisson regression models are shown in Figure 1.

From Figure 1, it can be seen that $A_h$-Trans-GLM outperforms naive-Lasso for most combinations of $h$ and $K$. As more and more source data become available, the performance of $A_h$-Trans-GLM improves significantly. This matches our theoretical analysis because the $\ell_2$-estimation error bounds in Theorems 1 and 3 become sharper as $n_{A_h}$ grows. When $h$ increases, the performance of $A_h$-Trans-GLM becomes worse.

We also apply the inference Algorithm 3 with $A_h$ 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 $\beta$, and report three evaluation metrics in Figure 2 when $h = 20$ under different $K_{A_h}$: (i) the average estimation error of $\Theta_j$ over variables in the signal set $S$ and noise set $S'$ (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 $S'$; (iii) the average CI length over $j \in$ signal set $S$ and noise set $S'$. Note that there is no explicit formula of $\Theta_j$ for logistic and Poisson regression models. Here we approximated it through 5 $\times$ $10^6$ Monte Carlo simulations. Notice that the average estimation error of $A_h$-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 $S'$ 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 $A_h$-Trans-GLM can achieve approximately 95% level when $K$ is large on both $S$ and $S'$. Finally, the results of average CI length reveal that the CIs obtained by $A_h$-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 $\Theta_j$ which leads to too narrow CIs to cover the true coefficients. And $A_h$-Trans-GLM offers a more accurate estimate of $\Theta_j$ which results in wider CIs.

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

4.1.2. Transfer Learning When $A_h$ is Unknown

Different from the previous section, now we fix the total number of sources as $K = 10$. There are two types of sources, which belong to either $A_h$ or $A_h^c$. Sources from $A_h$ have similar coefficients to the target one, while the coefficients of sources outside $A_h$ can be quite different. Intuitively, using more sources from $A_h$ benefits the estimation of the target coefficient. But in
Figure 1. The average $\ell_2$-estimation error of $A_h$-Trans-GLM and naive-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.

Figure 2. Three evaluation metrics of Algorithm 3 with $A_h$ (we denote it as $A_h$-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.
practice, \( A_c \) 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 \( n_t = 200 \) 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 \( R_p^{(k)} \) denotes \( p \) independent Rademacher variables and \( R_p^{(k')} \) are independent for any \( k \neq k' \). Consider \( h = 20 \) and 40. For any source data \( k \in \mathcal{A}_h \), we set \( w^{(k)} = \beta + (h/p) R_p^{(k)} \).

For linear and logistic regression models, predictors from target \( x_i^{(0)} \sim N(0, \Sigma) \) with \( \Sigma = [\Sigma_{ij}]_{p \times p} \) where \( \Sigma_{ij} = 0.9^{j-i} \), 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 \( \pm 0.5 \).

To generate the coefficient \( w^{(k)} \) for \( k \not\in \mathcal{A}_h \), we randomly generate \( S^{(k)} \) of size \( s \) from \( \{2s+1, \ldots, p\} \). Then, the \( j \)th component of coefficient \( w^{(k)} \) is set to be

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

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. \( A_h \)-Trans-GLM and Pooled-Trans-GLM represent Algorithm 1 on source data in \( A_h \) and target data or all sources and target data, respectively. Trans-GLM runs Algorithm 2 by first identifying the informative source set \( \mathcal{A} \), then applying Algorithm 1 to fit the model on sources in \( \mathcal{A} \). We vary the values of \( K_{A_h} \) and \( h \), and repeat simulations in each setting 200 times. The average \( \ell_2 \)-estimation errors are summarized in Figure 3.

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

### 4.2. A Real-Data Study

In this section, we study the 2020 U.S. 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 [github.com/tonmcg/US_County_Level_Election_Results_08-20](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.
Figure 4. The transferability between different states for Trans-GLM.

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 preprocessing, there are 3111 counties and 1081 predictors in the final data, belonging to 49 U.S. 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 \(\hat{A}\). 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
which can be seen as an extension to Bastani (2021) and Li, high-dimensional GLM under a transfer learning framework, In this work, we study the GLM transfer learning problem.

5. Discussions

like group Lasso (Yuan and Lin 2006) or Lasso with hierarchy that besides the classical Lasso penalty, there are other variants main effects and their interactions, one reviewer pointed out the details there. Furthermore, since we have considered all 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 $\ell_1/\ell_2$ estimation error rate when the homogeneity assumption (Assumption 4) holds. Without homogeneity of predictors between target and source, only suboptimal rates are obtained. This problem exists in the line of most existing high-dimensional transfer learning research (Li, Cai, and Li 2020, 2021; Bastani 2021). 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.

S.1.3.4 of supplementary materials. 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, Taylor, and Tibshirani 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 article to study high-dimensional GLM under a transfer learning framework, which can be seen as an extension to Bastani (2021) and Li, Cai, and Li (2021). We propose GLM transfer learning algorithms, and derive bounds for $\ell_1/\ell_2$-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 $\ell_1/\ell_2$ estimation error rate when the homogeneity assumption (Assumption 4) holds. Without homogeneity of predictors between target and source, only suboptimal rates are obtained. This problem exists in the line of most existing high-dimensional transfer learning research (Li, Cai, and Li 2020, 2021; Bastani 2021). 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 Materials

We summarize the additional simulation and real data analysis results, more theoretical details as well as all the proofs in the supplements.

Acknowledgments

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

Funding

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.

ORCID

Yang Feng http://orcid.org/0000-0001-7746-7598

References

Bastani, H. (2021), “Predicting with Proxies: Transfer Learning in High Dimension,” Management Science, 67, 2964–2984. [2684,2685,2686,2688,2696]

Bien, J., Taylor, J., and Tibshirani, R. (2013), “A Lasso for Hierarchical Interactions,” The Annals of Statistics, 41, 1111–1141. [2696]

Cai, T. T., and Wei, H. (2021), “Transfer Learning for Nonparametric Classification: Minimax Rate and Adaptive Classifier,” The Annals of Statistics, 49, 100–128. [2684]

Chen, A., Owen, A. B., Shi, M. (2015), “Data Enriched Linear Regression,” Electronic Journal of Statistics, 9, 1078–1112. [2685]

Gross, S. M., and Tibshirani, R. (2016), “Data Shared Lasso: A Novel Tool to Discover Uplift,” Computational Statistics & Data Analysis, 101, 226–235. [2685]

Hajiramezani, E., and Zamani, S. (2018), “Bayesian Multi-Domain Learning for Cancer Subtype Discovery from Next-Generation Sequencing Count Data,” in Advances in Neural Information Processing Systems 31 (NIPS 2018), 31. [2684]

Hanneke, S., and Kpotufe, S. (2020a), “A no-free-lunch Theorem for Multi-task Learning,” arXiv preprint arXiv:2006.15785. [2685]

——— (2020b), “On the Value of Target Data in Transfer Learning,” arXiv preprint arXiv:2002.04747. [2685]

Li, S., Cai, T. T., and Li, H. (2020), “Transfer Learning in Large-Scale Gaussian Graphical Models with False Discovery Rate Control,” arXiv preprint arXiv:2010.11037. [2684,2696]

——— (2021), “Transfer Learning for High-Dimensional Linear Regression: Prediction, Estimation and Minimax Optimality,” Journal of the Royal Statistical Society, Series B (to appear). [2684,2685,2686,2688,2689,2696]
McCullagh, P., and Nelder, J. A. (1989), *Generalized Linear Models* (Vol. 37), Boca Raton, FL: CRC Press. [2685]

Meinshausen, N., and Bühlmann, P. (2006), “High-Dimensional Graphs and Variable Selection with the Lasso,” *The Annals of Statistics*, 34, 1436–1462. [2687]

Negahban, S., Yu, B., Wainwright, M. J., and Ravikumar, P. K. (2009), "A Unified Framework for High-Dimensional Analysis of m-estimators with Decomposable Regularizers," in *Advances in Neural Information Processing Systems*, pp. 1348–1356. Citeseer. [2688]

Ollier, E., and Viallon, V. (2017), “Regression Modelling on Stratified Data with the Lasso,” *Biometrika*, 104, 83–96. [2685]

Pan, S. J., and Yang, Q. (2009), ”A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, 22, 1345–1359. [2684,2686]

Reeve, H. W., Cannings, T. I., and Samworth, R. J. (2021), ”Adaptive Transfer Learning,” *The Annals of Statistics*, 49, 3618–3649. [2684]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288. [2684]

Torrey, L., and Shavlik, J. (2010), ”Transfer Learning,” in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, eds. E. S. Olivas, J .D. M. Guerrero, M. M. Sober, J. R. M. Benedito, A. J. S. López, pp. 242–264. Hershey, PA: IGI Global. [2684,2685,2686]

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, 1166–1202. [2685,2687,2691,2692]

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)*, pp. 617–626. IEEE. [2684]

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016), ”A Survey of Transfer Learning,” *Journal of Big Data*, 3, 1–40. [2684,2686]

Yuan, M., and Lin, Y. (2006), ”Model Selection and Estimation in Regression with Grouped Variables,” *Journal of the Royal Statistical Society, Series B*, 68, 49–67. [2696]

Zheng, C., Dasgupta, S., Xie, Y., Haris, A., and Chen, Y. Q. (2019), ”On Data Enriched Logistic Regression,” arXiv preprint arXiv:1911.06380. [2685]