Predicting with Proxies: Transfer Learning in High Dimension

Hamsa Bastani
Wharton School, Operations Information and Decisions, hamsab@wharton.upenn.edu

Predictive analytics is increasingly used to guide decision-making in many applications. However, in practice, we often have limited data on the true predictive task of interest, and must instead rely on more abundant data on a closely-related *proxy* predictive task. For example, e-commerce platforms use abundant customer click data (proxy) to make product recommendations rather than the relatively sparse customer purchase data (true outcome of interest); alternatively, hospitals often rely on medical risk scores trained on a different patient population (proxy) rather than their own patient population (true cohort of interest) to assign interventions. Yet, not accounting for the bias in the proxy can lead to sub-optimal decisions. Using real datasets, we find that this bias can often be captured by a sparse function of the features. Thus, we propose a novel two-step estimator that uses techniques from high-dimensional statistics to efficiently *combine* a large amount of proxy data and a small amount of true data. We prove upper bounds on the error of our proposed estimator and lower bounds on several heuristics used by data scientists; in particular, our proposed estimator can achieve the same accuracy with exponentially less true data (in the number of features d). Our proof relies on a new LASSO tail inequality for approximately sparse vectors. Finally, we demonstrate the effectiveness of our approach on e-commerce and healthcare datasets; in both cases, we achieve significantly better predictive accuracy as well as managerial insights into the nature of the bias in the proxy data.

Key words: proxies, transfer learning, sparsity, high-dimensional statistics, LASSO

1. Introduction

Decision-makers increasingly use machine learning and predictive analytics to inform consequential decisions. However, a pervasive problem that occurs in practice is the limited quantity of labeled data available in the desired setting. Building accurate predictive models requires significant quantities of labeled data, but large datasets may be costly or infeasible to obtain for the predictive task of interest. A common solution to this challenge is to rely on a *proxy* — a closely-related predictive task — for which abundant data is already available. The decision-maker then builds and deploys a model predicting the proxy instead of the true task. To illustrate, consider the following two examples from revenue management and healthcare respectively:

Example 1 (Recommendation Systems). A core business proposition for platforms (e.g., Expedia or Amazon) is to match customers with personalized product recommendations. The typical goal is to maximize the probability of a customer purchase by recommending products that a

customer is most likely to purchase, based on past transaction data and customer purchase histories. Unfortunately, most platforms have sparse data on customer purchases (the true outcome they wish to predict) for a particular product, but significantly more data on customer *clicks* (a proxy outcome). Clicks are a common proxy for purchases, since one may assume that customers will not click on a product without some intent to purchase. Consequently, platforms often recommend products with high predicted click-through rates rather than high predicted purchase rates.

Example 2 (Medical Risk Scoring). Many hospitals are interested in identifying patients who have high risk for some adverse event (e.g., diabetes, stroke) in order to target preventative interventions. This involves using past electronic medical records to train a patient *risk score*, i.e., predict which patients are likely to get a positive diagnosis for the adverse event based on data from prior visits. However, small hospitals have limited data since their patient cohorts (true population of interest) are not sizable enough to have had a large number of adverse events. Instead, they adopt a published risk score trained on data from a larger hospital's patient cohort (proxy population). There are concerns that a predictive model trained at one hospital may not directly apply to a different hospital, since there are differences in physician behavior, patient populations, etc. Yet, one may assume that the large hospital's risk score is a good proxy for the small hospital's risk score, since the target of interest (the adverse event) is the same in both models.

There are numerous other examples of the use of proxies in practice. In Section 1.1, we overview the pervasiveness of proxies in healthcare and revenue management.

However, the use of proxies has clear drawbacks: the proxy and true predictive models may not be the same, and any bias between the two tasks will affect the predictive performance of the model. Consider Example 1 on recommendation systems. In Section 5.2, we use personalized hotel recommendation data from Expedia to demonstrate a systematic bias between clicks (proxy outcome) and purchases (true outcome). In particular, we find that the price of the recommendation negatively impacts purchases far more than clicks. Intuitively, a customer may not mind browsing expensive travel products, but is unlikely to make an expensive purchase. Thus, using predicted click-through rates alone (proxies) to make recommendations could result in overly expensive recommendations, thereby hurting purchase rates. Next, consider Example 2 on medical risk scores. In Section 5.3, we use electronic medical record data across several healthcare providers to demonstrate a systematic bias between a diabetes risk predictor trained on patient data from a large external hospital (proxy cohort) and a risk predictor trained on patient data from the small target hospital (true cohort). In particular, we find differences in physician diagnosing behavior (e.g., some physicians are more inclined than others to ask patients to fast in order to diagnose impaired fasting glucose) and how patient chart data is encoded in the medical record (e.g., obesity is recorded as a diagnosis more often in some hospitals despite patients having similar BMIs). As a result, features that are

highly predictive in one hospital may not be predictive in another hospital, thereby hurting the performance of a borrowed (proxy) risk predictor at the target hospital.

We refer to data from the proxy and true predictive tasks as proxy and *gold* data respectively. Analogously, we refer to estimators trained on proxy and gold data alone as the proxy and gold estimators respectively. Both estimators seek to predict outcomes on the true predictive task. From a statistical perspective, the gold estimator is unbiased but has high variance due to its limited sample size. On the other hand, the proxy estimator has low variance due to its large sample size, but may have a significant bias due to systematic differences between the true and proxy predictive tasks. Predictive accuracy is composed of both bias and variance. Thus, when we have a good proxy (the bias is not too large), the proxy estimator can be a much more accurate predictive model than the gold estimator, explaining the wide use of proxies in practice.

An immediate question is: can we *combine* proxy and gold data to achieve a better bias-variance tradeoff and improve predictive accuracy? In many of these settings, we have access to (or could collect) information from *both* predictive tasks, i.e., we typically have a large amount of proxy data *and* a small amount of true data. For instance, platforms observe both clicks and purchases; the target hospital has access to both the published proxy estimator and basic summary statistics from an external hospital, as well as its own patient data. Thus, we have the opportunity to improve prediction by combining these data sources. Conversations with professional data scientists indicate two popular heuristics: (i) model averaging over the gold and proxy estimators, and (ii) training a model on proxy and gold data simultaneously[1], with a larger weight for gold observations. However, there is little understanding of whether and by how much these heuristics can improve predictive performance. Indeed, we prove lower bounds that both model averaging and weighted loss functions can only improve estimation error by at most a constant factor (beyond the naive proxy and gold estimators discussed earlier). Thus, neither approach can significantly improve estimation error.

Ideally, we would use the gold data to *de-bias* the proxy estimator (which already has low variance); this would hopefully yield an estimator with lower bias while maintaining low variance. However, estimating the bias is challenging, as we have extremely limited gold data. In general, estimating the bias from gold data can be harder than directly estimating the true predictive model from gold data. Thus, we clearly need to impose additional structure to make progress.

Our key insight is that the bias between the true and proxy predictive tasks may often be well modeled by a *sparse* function of the observed features. We argue that there is often some (a priori unknown) underlying mechanism that systematically affects a subset of the features, creating a bias between the true and proxy predictive tasks. When this is the case, we can successfully estimate the

[1] One disadvantage of the weighted loss function is that it requires both proxy and gold data to be available together at the time of training. This may not be possible in settings such as healthcare, where data is sensitive.

bias using high-dimensional techniques that exploit sparsity. To illustrate, we return to Examples 1 and 2. In the first example on hotel recommendations, we find on Expedia data that customers tend to click on more expensive products than they are willing to purchase. This creates a bias between the proxy and true predictive tasks that can be captured by the price feature alone. However, as we show in Fig. 2 in Section 5.2, the two predictive tasks appear remarkably similar otherwise. In particular, the difference of the proxy and gold estimators on Expedia data is very sparse (nearly all coefficients are negligible with the notable exception of the price coefficient). Similarly, in the second example on diabetes risk prediction, we find on patient data that physicians/coders at different hospitals sometimes diagnose/record different conditions in the electronic medical record. However, the majority of patient data is similarly diagnosed and recorded across hospitals (motivating the common practice of borrowing risk predictors from other hospitals). This creates a bias between the proxy and true predictive tasks that can be captured by the few features corresponding only to the subset of diagnoses where differences arise.

Importantly, in both examples, the proxy and gold estimators themselves are *not* sparse. Thus, we cannot exploit this structure by directly applying high-dimensional techniques to proxy or gold data separately. Rather, we must efficiently combine proxy and gold data, while exploiting the sparse structure of the bias *between* the two predictive tasks. Our lower bounds show that popular heuristics (model averaging and weighted loss functions) fail to leverage sparse structure even when it is present, and can still only improve predictive accuracy by at most a constant factor.

We propose a new two-step joint estimator that successfully leverages sparse structure in the bias term to achieve a much stronger improvement in predictive accuracy. In particular, our proposed estimator can achieve the same accuracy with *exponentially* less gold data (in the number of features d). Intuitively, instead of using the limited gold data directly for estimating the predictive model, our estimator uses gold data to efficiently de-bias the proxy estimator. In fact, when gold data is very limited, the availability of proxy data is critical to extracting value from the gold data. Our proof relies on a new LASSO tail inequality for approximately sparse vectors, which may be of independent interest. It is worth noting that our estimator does not simultaneously require both proxy and gold data at training time; this is an important feature in settings such as healthcare, where data from different sources cannot be combined due to regulatory constraints. We demonstrate the effectiveness of our estimator on both Expedia hotel recommendation (Example 1) and diabetes risk prediction (Example 2). In both cases, we achieve significantly better predictive accuracy, as well as managerial insights into the nature of the bias in the proxy data.

1.1. Pervasiveness of Proxies

Proxies are especially pervasive in healthcare, where patient covariates and response variables must be derived from electronic medical records (EMRs), which are inevitably biased by the data

collection process. One common issue is *censoring*: we only observe a diagnosis in the EMR *if* the patient visits the healthcare provider. Thus, the recorded diagnosis code (often used as the response variable) is in fact a proxy for the patient's true outcome (which may or may not have been recorded). Mullainathan and Obermeyer (2017) and Obermeyer and Lee (2017) demonstrate that this proxy can result in misleading predictive models, arising from systematic biases in the types of patients who frequently visit the healthcare provider. One could collect more reliable (true) outcome data by surveying patients, but this is costly and only scales to a small cohort of patients. Another form of censoring is *omitted variable bias*: important factors (e.g., physician counseling or a patient's proactiveness towards their own health) are not explicitly recorded in the medical record. Bastani et al. (2017) show that omitted variable bias arising from unrecorded physician interventions can lead to misleading predictive models trained on EMR data. Again, more reliable (gold) data can be collected by hand-labeling patient observations based on physician or nurse notes in the medical chart, but as before, this is costly and unscalable. Recently, researchers have drawn attention to *human bias*: patient data is collected and recorded by hospital staff (e.g., physicians, medical coders), who may themselves be biased (Ahsen et al. 2018). This is exemplified in our case study (Section 5.3), where we find that medical coders record the obesity diagnosis code in the EMR at very different rates even when patient BMIs are similar. Finally, the specific outcomes of interest may be *too rare* or have high variance. For example, in healthcare pay-for-performance contracts, Medicare uses 30-day hospital readmissions rates as proxies for hospital quality of care, which may be better captured by rarer outcomes such as never events or 30-day patient mortality rates (CMS 2018, Axon and Williams 2011, Milstein 2009).

Proxies are also pervasive in marketing and revenue management. Online platforms allow us to observe fine-grained customer behaviors, including page views, clicks, cart-adds, and eventually purchases. While purchases may be the final outcome of interest, these intermediate (and more abundant) observations serve as valuable proxies. For example, Farias and Li (2017) use a variety of customer actions as proxies for predicting a customer's affinity for a song in a music streaming service. This is also evidenced in our case study (Section 5.2), where customer clicks can signal the likelihood of customer hotel purchases. With modern technology, companies can also observe customers' offline behavior, including store visits (using mobile WiFi signal tracking, e.g., see Zhang et al. 2018 for Alibaba case study) and real-time product browsing (using store security cameras, e.g., see Brynjolfsson et al. 2013 for American Apparel case study). Thus, different channels of customer behavior can inform predictive analytics. For example, Dzyabura et al. (2018) use online customer behaviors as proxies for predicting offline customer preferences. Finally, new product introduction can benefit from proxies. For example, Baardman et al. (2017) use demand for related products as proxies for predicting demand for a new product.

1.2. Other Related Work

Our problem can be viewed as an instance of multitask learning, or more specifically, *transfer learning*. Multitask learning combines data from multiple related predictive tasks to train similar predictive models for each task. It does this by using a *shared representation* across tasks (Caruana 1997). Such representations typically include variable selection (i.e., enforce the same feature support for all tasks in linear or logistic regression, Jalali et al. 2010, Meier et al. 2008), kernel choice (i.e., use the same kernel for all tasks in kernel regression, Caruana 1997), or intermediate neural net representations (i.e., use the same weights for intermediate layers for all tasks in deep learning, Collobert and Weston 2008). Transfer learning specifically focuses on learning a single new task by transferring knowledge from a related task that has already been learned (see Pan et al. 2010 for a survey). We share a similar goal: since we have many proxy samples, we can easily learn a high-performing predictive model for the proxy task, but we wish to transfer this knowledge to the (related) gold task for which we have very limited labeled data. However, our proxy and gold predictive models already have a shared representation in the variable selection sense; in particular, we use the same features (all of which are typically relevant) for both prediction tasks.

We note that the tasks considered in the multitask and transfer learning literature are typically far more disparate than the class of proxy problems we have identified in this paper thus far. For instance, Caruana (1997) gives the example of simultaneously training neural network outputs to recognize different object properties (outlines, shapes, textures, reflections, shadows, text, orientation, etc.). Bayati et al. (2018) simultaneously train logistic regressions predicting disparate diseases (heart failure, diabetes, dementia, cancer, pulmonary disorder, etc.). While these tasks are indeed related, they are not close substitutes for each other. In contrast, the proxy predictive task *is* a close substitute for the true predictive task, to the point that practitioners may even ignore gold data and train their models purely on proxy data. In this class of problems, we can impose significantly more structure beyond merely a shared representation.

Our key insight is that the bias between the proxy and gold predictive tasks can be modeled as a sparse function. We argue that there is often some (a priori unknown) underlying mechanism that systematically affects a subset of the features, creating a bias between the true and proxy predictive tasks. When this is the case, we can successfully estimate the bias using high-dimensional techniques that exploit sparsity.

Bayesian approaches have been proposed for similar problems. For instance, Dzyabura et al. (2018) use a Bayesian prior relating customers' online preferences (proxies) and offline purchase behavior (true outcome of interest). Raina et al. (2006) propose a method for constructing priors in such settings using semidefinite programming on data from related tasks. These approaches do not come with theoretical convergence guarantees. A frequentist interpretation of their approach

is akin to ridge regression, which is one of our baselines; we prove that ridge regression cannot take advantage of sparse structure when present, and thus, cannot significantly improve improve estimation error over the naive proxy or gold estimators. Relatedly, Farias and Li (2017) link multiple low-rank collaborative filtering problems by imposing structure across their latent feature representations; however, the primary focus in their work is on low-rank matrix completion settings without features, whereas our focus is on classical regression problems.

We use techniques from the high-dimensional statistics literature to prove convergence properties about our two-step estimator. The second step of our estimator uses a LASSO regression (Chen et al. 1995, Tibshirani 1996), which helps us recover the bias term using far fewer samples than traditional statistical models by exploiting sparsity (Candes and Tao 2007, Bickel et al. 2009, Negahban et al. 2009). A key challenge in our proof is that the vector we wish to recover in the second stage is not perfectly sparse; rather, it is the sum of a sparse vector and residual noise from the first stage of our estimator. Existing work has studied convergence of LASSO for approximately sparse vectors (Bühlmann and Van De Geer 2011, Belloni et al. 2012), while making little to no assumptions on the nature of the approximation; we extend this theory to prove a tighter but structure-dependent tail inequality that is appropriate for our setting. As a result, we show that the error of our joint estimator cleanly decomposes into a term that is proportional to the variance of our proxy estimator (which is small in practice), and a term that recovers the classical error rate of the LASSO estimator. Thus, when we have many proxy observations, we require exponentially fewer gold observations to achieve a nontrivial estimation error than would be required if we did not have any proxy data. Our two-stage estimator is related in spirit to other high-dimensional two-stage estimators (e.g., Belloni et al. 2014, 2012). While these papers focus on treatment effect estimation after variable selection on features or instrumental variables, our work focuses on transfer learning from a proxy predictive task to a new predictive task with limited labeled data.

1.3. Contributions

We highlight our main contributions below:

1. *Problem Formulation:* We formulate the proxy problem as two classical regression tasks; the proxy task has abundant data, while the actual (gold) task of interest has limited data. Motivated by real datasets, we model the bias between the two tasks as a sparse function of the features.

2. *Theory:* We propose a new two-step estimator that efficiently combines proxy and gold data to exploit sparsity in the bias term. Our estimator provably achieves the same accuracy as popular heuristics (e.g., model averaging or weighted loss functions) with exponentially less gold data (in the number of features d). Our proof relies on a new tail inequality on the convergence of LASSO for approximately sparse vectors, which may be of independent interest.

3. *Case Studies:* We demonstrate the effectiveness of our approach on e-commerce and healthcare datasets. In both cases, we achieve significantly better predictive accuracy as well as managerial insights into the nature of the bias in the proxy data.

2. Problem Formulation

Preliminaries: For any integer n, let $[n]$ denote the set $\{1,...,n\}$. Consider an observation with feature vector $\mathbf{x} \in \mathbb{R}^d$. As discussed earlier, the gold and predictive tasks are different. Let the gold and proxy responses be given by the following linear data-generating processes respectively (we will discuss nonlinear parametric models in Section 4.5):

$$y_{gold} = \mathbf{x}^\top \beta^*_{gold} + \varepsilon_{gold}\,,$$
$$y_{proxy} = \mathbf{x}^\top \beta^*_{proxy} + \varepsilon_{proxy}\,,$$

where $\beta^*_{gold}, \beta^*_{proxy} \in \mathbb{R}^d$ are unknown regression parameters, and the noise $\varepsilon_{gold}, \varepsilon_{proxy}$ are each vectors of independent subgaussian variables with parameters σ_{gold} and σ_{proxy} respectively (see Definition 1 below). We do not impose that ε_{gold} is independent of ε_{proxy}; for example, $(\varepsilon^{(i)}_{gold}, \varepsilon^{(i)}_{proxy})$ can be arbitrarily pairwise correlated[2] for any i.

Definition 1. A random variable $z \in \mathbb{R}$ is σ-subgaussian if $\mathbb{E}[e^{tz}] \le e^{\sigma^2 t^2/2}$ for every $t \in \mathbb{R}$.

This definition implies $\mathbb{E}[z] = 0$ and $\mathrm{Var}[z] \le \sigma^2$. Many classical distributions are subgaussian; typical examples include any bounded, centered distribution, or the normal distribution. Note that the errors need not be identically distributed.

Our goal is to estimate β^*_{gold} accurately in order to make good decisions for new observations with respect to their true predicted outcomes. In a typical regression problem, the gold data would suffice. However, we often have very limited gold data, leading to high-variance erroneous estimates. This can be either because n_{gold} is small (Example 2) or σ_{gold} is large (Example 1). Thus, we can benefit by utilizing information from proxy data, even if this data is biased.

Decision-makers employ proxy data because the proxy predictive task is closely related to the true predictive task. In other words, $\beta^*_{gold} \approx \beta^*_{proxy}$. To model the relationship between the true and proxy predictive tasks, we write

$$\beta^*_{gold} = \beta^*_{proxy} + \delta^*\,,$$

where δ^* captures the proxy estimator's bias.

Motivated by our earlier discussion, we posit that the bias is *sparse*. In particular, let $\|\delta^*\|_0 = s$, which implies that the bias of the proxy estimator only depends on s out of the d covariates. This

[2] In Example 1, a customer's click (proxy) and purchase (gold) responses may have correlated customer-specific noise.

constraint is always satisfied when $s = d$, but we will prove that our estimator of β^*_{gold} has much stronger performance guarantees when $s \ll d$.

Data: We are given two (possibly overlapping) cohorts. We have n_{gold} observations in our gold dataset: let $\mathbf{X}_{gold} \in \mathbb{R}^{n_{gold} \times d}$ be the gold design matrix (whose rows are observations from the gold cohort), and $Y_{gold} \in \mathbb{R}^{n_{gold}}$ be the corresponding vector of responses. Analogously, we have n_{proxy} observations in our proxy dataset: let $\mathbf{X}_{proxy} \in \mathbb{R}^{n_{proxy} \times d}$ be the proxy design matrix (whose rows are observations from the proxy cohort), and $Y_{proxy} \in \mathbb{R}^{n_{proxy}}$ be the corresponding vector of responses. Typically $n_{gold} \ll n_{proxy}$ or $\sigma_{gold} \gg \sigma_{proxy}$, necessitating the use of proxy data. Without loss of generality, we impose that both design matrices have been standardized, i.e.,

$$\left\|\mathbf{X}^{(r)}_{gold}\right\|_2^2 = n_{gold} \quad \text{and} \quad \left\|\mathbf{X}^{(r)}_{proxy}\right\|_2^2 = n_{proxy}\,,$$

for each column $r \in [d]$. It is standard good practice to normalize features in this way when using regularized regression, so that the regression parameters are appropriately scaled in the regularization term (see, e.g., Friedman et al. 2001). We further define the $d \times d$ gold and proxy *sample covariance matrices*

$$\Sigma_{gold} = \frac{1}{n_{gold}} \mathbf{X}^\top_{gold} \mathbf{X}_{gold} \quad \text{and} \quad \Sigma_{proxy} = \frac{1}{n_{proxy}} \mathbf{X}^\top_{proxy} \mathbf{X}_{proxy}\,.$$

Our standardization of the design matrices implies that $\operatorname{diag}(\Sigma_{gold}) = \operatorname{diag}(\Sigma_{proxy}) = \mathbf{1}_{d\times 1}$.

Evaluation: We define the parameter estimation error of a given estimator $\hat{\beta}$ relative to the true parameter β^*_{gold} as

$$R(\hat{\beta}, \beta^*_{gold}) = \sup_{\mathcal{S}} \; \mathbb{E}\left[\left\|\hat{\beta} - \beta^*_{gold}\right\|_1\right]\,,$$

where $\mathcal{S} = \left\{\mathbf{X}_{gold}, \mathbf{X}_{proxy}, \beta^*_{gold}, \delta^*\right\}$ is the set of feasible problem parameters[3] (i.e., satisfying the assumptions given in the problem formulation and Section 2.1), and the expectation is taken with respect to the noise terms ε_{gold} and ε_{proxy}. Note that a bound on R implies a bound on the expected out-of-sample prediction error for any new bounded observation $x \in \mathbb{R}^d$, i.e., by Hölder's inequality,

$$\mathbb{E}\left[\left|x^\top \hat{\beta} - x^\top \beta^*_{gold}\right|\right] \;\le\; \mathbb{E}\left[\left\|\hat{\beta} - \beta^*_{gold}\right\|_1\right] \cdot \|x\|_\infty \;\le\; R(\hat{\beta}, \beta^*_{gold}) \cdot \|x\|_\infty\,.$$

2.1. Assumptions

ASSUMPTION 1 **(Bounded)**. *There exists some $b \in \mathbb{R}$ such that $\left\|\beta^*_{gold}\right\|_1 \le b$.*

Our first assumption states that our regression parameters are bounded by some constant[4]. This is a standard assumption in the statistical literature.

[3] Note that β^*_{proxy} is implicitly defined in $\mathcal{S}$ as $\beta^*_{gold} - \delta^*$.

[4] Note that this does not imply that β^*_{gold} is sparse, e.g., $u = \frac{1}{d}\mathbf{1} \in \mathbb{R}^d$ satisfies $\|u\|_1 = 1$ but $\|u\|_0 = d$.

ASSUMPTION 2 **(Positive-Definite)**. *The proxy sample covariance matrix Σ_{proxy} is positive-definite. In other words, the minimum eigenvalue of Σ_{proxy} is $\psi > 0$.*

Our second assumption is also standard, and ensures that β^*_{proxy} is identifiable from the proxy data $(\mathbf{X}_{proxy}, Y_{proxy})$. This is a mild assumption since n_{proxy} is large. In contrast, we allow that β^*_{gold} may not be identifiable from the gold data $(\mathbf{X}_{gold}, Y_{gold})$, since n_{gold} is small and the resulting sample covariance matrix Σ_{gold} may not be positive-definite.

The last assumption on the *compatibility condition* arises from the theory of high-dimensional statistics (Candes and Tao 2007, Bickel et al. 2009, Bühlmann and Van De Geer 2011). We will require a few definitions before stating the assumption.

An *index set* is a set $S \subset [d]$. For any vector $u \in \mathbb{R}^d$, let $u_S \in \mathbb{R}^d$ be the vector obtained by setting the elements of u that are not in S to zero. Then, the i^{th} element of u_S is $u_S^{(i)} = u^{(i)} \cdot \mathbb{1}[i \in S]$. Furthermore, let S^c denote the complement of S. Then, $S \cup S^c = [d]$ and $S \cap S^c = \varnothing$.

The *support* for any vector $u \in \mathbb{R}^d$, denoted $supp(u) \subset [d]$, is the set of indices corresponding to nonzero entries of u. Thus, $supp(u)$ is the smallest set that satisfies $u_{supp(u)} = u$.

We now define the compatibility condition:

DEFINITION 2 (COMPATIBILITY CONDITION). The compatibility condition is met for the index set $S \subseteq [d]$ and the matrix $\Sigma \in \mathbb{R}^{d \times d}$ if there exists $\phi > 0$ such that, for all $u \in \mathbb{R}^d$ satisfying $\|u_{S^c}\|_1 \leq 3\|u_S\|_1$, it holds that

$$\|u_S\|_1^2 \leq \frac{|S|}{\phi^2} \left(u^T \Sigma u\right) .$$

ASSUMPTION 3 **(Compatibility Condition)**. *The compatibility condition (Definition 2) is met for the index set $S = supp(\delta^*)$ and gold sample covariance matrix Σ_{gold} with constant $\phi > 0$.*

Our third assumption is critical to ensure that the bias term δ^* is identifiable, even if $n_{gold} < d$. This assumption (or the related restricted eigenvalue condition) is standard in the literature to ensure the convergence of high-dimensional estimators such as the Dantzig selector or LASSO (Candes and Tao 2007, Bickel et al. 2009, Bühlmann and Van De Geer 2011).

It is worth noting that Assumption 3 is always satisfied if Σ_{gold} is positive-definite. In particular, letting $\zeta > 0$ be the minimum eigenvalue of Σ_{gold}, it can be easily verified that the compatibility condition holds with constant $\phi_0 = \sqrt{\zeta}$ for *any* index set. Thus, the compatibility condition is strictly weaker than the requirement that Σ_{gold} be positive-definite. For example, the compatibility condition allows for collinearity in features that are outside the index set S, which can occur often in high-dimensional settings when $|S| = s \ll d$ (Bühlmann and Van De Geer 2011). Thus, even when β^*_{gold} is not identifiable, we may be able to identify the bias δ^* by exploiting sparsity.

3. Baseline Estimators

We begin by describing four commonly used baseline estimators. These include naive estimators trained only on gold or proxy data, as well as two popular heuristics (model averaging and weighted loss functions). We prove corresponding lower bounds on their parameter estimation error $R(\cdot, \beta^*_{gold})$ with respect to the true parameter β^*_{gold}.

3.1. OLS/Ridge Estimator on Gold Data

One common approach is to ignore proxy data and simply use the gold data (the most appropriate data) to construct the best possible predictor. Since we have a linear model, the ordinary least squares (OLS) estimator is the most obvious choice: it is the minimum variance unbiased estimator.

However, it is well known that introducing bias can be beneficial in data-poor environments. In other words, since we have very few gold samples (n_{gold} is small), we may wish to consider the regularized ridge estimator (Friedman et al. 2001):

$$\hat{\beta}^{ridge}_{gold}(\lambda) \;=\; \arg\min_{\beta} \left\{ \frac{1}{n_{gold}} \left\| Y_{gold} - \mathbf{X}_{gold}\beta \right\|_2^2 + \lambda \left\| \beta \right\|_2^2 \right\},$$

where we introduce a regularization parameter $\lambda \geq 0$. Note that when the regularization parameter $\lambda = 0$, we recover the classical OLS estimator, i.e., $\hat{\beta}^{ridge}_{gold}(0) = \hat{\beta}^{OLS}_{gold}$.

THEOREM 1 **(Gold Estimator)**. *The parameter estimation error of the OLS estimator on gold data* $(\mathbf{X}_{gold}, Y_{gold})$ *is bounded below as follows:*

$$\begin{aligned} R\left(\hat{\beta}^{OLS}_{gold}, \beta^*_{gold}\right) \;&\geq\; d\sqrt{\frac{2\sigma^2_{gold}}{\pi n_{gold}}} \\ &=\; \mathcal{O}\left(\frac{d\sigma_{gold}}{\sqrt{n_{gold}}}\right). \end{aligned}$$

The parameter estimation error of the ridge estimator on gold data $(\mathbf{X}_{gold}, Y_{gold})$ *for any choice of the regularization parameter* $\lambda \geq 0$ *is bounded below as follows:*

$$\begin{aligned} \min_{\lambda \geq 0} \; R\left(\hat{\beta}^{ridge}_{gold}(\lambda), \beta^*_{gold}\right) \;&\geq\; \frac{d\sigma_{gold}/\sqrt{2\pi}}{b\sqrt{n_{gold}} + d\sigma_{gold}\sqrt{2/\pi}} \\ &=\; \mathcal{O}\left(\frac{d\sigma_{gold}}{\sqrt{n_{gold}} + d\sigma_{gold}}\right). \end{aligned}$$

The proof is given in Appendix A.1. Note that this result uses the optimal value of the regularization parameter λ to compute the lower bound on the parameter estimation error of the ridge estimator. In practice, the error will be larger since λ would be estimated through cross-validation.

Theorem 1 shows that when the number of gold samples is moderate (i.e., $n_{gold} \gg d\sigma_{gold}$), the ridge estimator recovers the OLS estimator's lower bound on the parameter estimation error $\mathcal{O}\left(d\sigma_{gold}/n_{gold}\right)$. However, when the number of gold samples is very small (i.e., $n_{gold} \lesssim d\sigma_{gold}$), the

ridge estimator achieves a constant lower bound on the parameter estimation error $\mathcal{O}(1)$. This is because the ridge estimator will predict $\hat{\beta}^{ridge}_{gold}(\lambda) = 0$ for very small values of n_{gold}, and since we have assumed that $\left\|\beta^*_{gold}\right\|_1 \leq b = \mathcal{O}(1)$, our parameter estimation error remains bounded.

3.2. OLS Estimator of Proxy Data

Another common approach is to ignore the gold data and simple use the proxy data to construct the best possible predictor. Since we have a linear model, the OLS estimator is the most obvious choice; note that we do not need regularization since we have many proxy samples (n_{proxy} is large). Thus, we consider:

$$\hat{\beta}_{proxy} \;=\; \arg\min_{\beta} \left\{ \frac{1}{n_{proxy}} \|y_{proxy} - \mathbf{X}_{proxy}\beta\|_2^2 \right\}.$$

THEOREM 2 **(Proxy Estimator)**. *The parameter estimation error of the OLS estimator on proxy data* $(\mathbf{X}_{proxy}, Y_{proxy})$ *is bounded below as follows:*

$$\begin{aligned} R\left(\hat{\beta}_{proxy}, \beta^*_{gold}\right) \;&\geq\; \max\left\{ \frac{1}{2}\left\|\delta^*\right\|_1, d\sqrt{\frac{\sigma^2_{proxy}}{2\pi n_{proxy}}} \right\} \\ &=\; \mathcal{O}\left(\left\|\delta^*\right\|_1 + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}} \right). \end{aligned}$$

The proof is given in Appendix A.2. Since n_{proxy} is large, the second term in the parameter estimation error $d\sigma_{proxy}/\sqrt{n_{proxy}}$ is small. Thus, the parameter estimation error of the proxy estimator is dominated by the bias term $\left\|\delta^*\right\|_1$. When the proxy is "good" or reasonably representative of the gold data, $\left\|\delta^*\right\|_1$ is small. In these cases, the proxy estimator is more accurate than the gold estimator, explaining the widespread use of the proxy estimator in practice even when (limited) gold data is available.

3.3. Model Averaging Estimator

One heuristic that is sometimes employed is to simply average the gold and proxy estimators:

$$\hat{\beta}_{avg}(\lambda) \;=\; (1-\lambda)\cdot\hat{\beta}^{OLS}_{gold} + \lambda\cdot\hat{\beta}_{proxy},$$

for some averaging parameter $\lambda \in [0,1]$. Note that $\lambda = 0$ recovers $\hat{\beta}^{OLS}_{gold}$ (the OLS estimator on gold data) and $\lambda = 1$ recovers $\hat{\beta}_{proxy}$ (the OLS estimator on proxy data).

THEOREM 3 **(Averaging Estimator)**. *The parameter estimation error of the averaging estimator on both gold and proxy data* $(\mathbf{X}_{gold}, Y_{gold}, \mathbf{X}_{proxy}, Y_{proxy})$ *is bounded below as follows:*

$$\begin{aligned} \min_{\lambda\in[0,1]} R\left(\hat{\beta}_{avg}(\lambda), \beta^*_{gold}\right) \;&\geq\; \min\left\{ \frac{d\sigma_{gold}}{3\sqrt{2\pi n_{gold}}},\; \frac{1}{6}\left\|\delta^*\right\|_1 + \frac{d\sigma_{proxy}}{3\sqrt{2\pi n_{proxy}}} \right\} \\ &=\; \mathcal{O}\left(\min\left\{ \frac{d\sigma_{gold}}{\sqrt{n_{gold}}},\; \left\|\delta^*\right\|_1 + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}} \right\} \right). \end{aligned}$$

The proof is given in Appendix A.3. Note that this result uses the optimal value of the averaging parameter λ to compute the lower bound on the parameter estimation error of the averaging estimator. In practice, the error will be larger since λ would be estimated through cross-validation.

Theorem 3 shows that the averaging estimator does not achieve more than a constant factor improvement over the best of the gold and proxy OLS estimators. In particular, the lower bound in Theorem 3 is exactly the minimum of the lower bounds of the gold OLS estimator (given in Theorem 1) and the proxy OLS estimator (given in Theorem 2) up to constant factors. Since the averaging estimator spans both the proxy and the gold estimators (depending on the choice of λ), it is to be expected that the best possible averaging estimator does at least as well as either of these two estimators; surprisingly, it does no better.

3.4. Weighted Loss Estimator

A more sophisticated heuristic used in practice is to perform a weighted regression that combines both datasets but assigns a higher weight to true outcomes. Consider:

$$\hat{\beta}_{weight}(\lambda) \;=\; \arg\min_{\beta} \left\{ \frac{1}{\lambda n_{gold} + n_{proxy}} \cdot \left(\lambda \|Y_{gold} - \mathbf{X}_{gold}\beta\|_2^2 + \|Y_{proxy} - \mathbf{X}_{proxy}\beta\|_2^2 \right) \right\},$$

for some weight $\lambda \in [0, \infty)$. Note that $\lambda = \infty$ recovers $\hat{\beta}_{gold}^{OLS}$ (the OLS estimator on gold data) and $\lambda = 0$ recovers $\hat{\beta}_{proxy}$ (the OLS estimator on proxy data).

THEOREM 4 **(Weighted Loss Estimator)**. *The parameter estimation error of the weighted estimator on both gold and proxy data* $(\mathbf{X}_{gold}, Y_{gold}, \mathbf{X}_{proxy}, Y_{proxy})$ *is bounded below as follows:*

$$\begin{aligned} \min_{\lambda \geq 0} \; R\left(\hat{\beta}_{weight}(\lambda), \beta_{gold}^{*}\right) \;&\geq\; \min\left\{ \frac{d\sigma_{gold}}{3\sqrt{2\pi n_{gold}}},\; \frac{1}{6}\left\|\delta^{*}\right\|_1 + \frac{d\sigma_{proxy}}{3\sqrt{2\pi n_{proxy}}} \right\} \\ &=\; \mathcal{O}\left(\min\left\{ \frac{d\sigma_{gold}}{\sqrt{n_{gold}}},\; \left\|\delta^{*}\right\|_1 + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}} \right\} \right). \end{aligned}$$

The proof is given in Appendix A.4. Note that this result uses the optimal value of the weighting parameter λ to compute the lower bound on the parameter estimation error of the weighted loss estimator. In practice, the error will be larger since λ would be estimated through cross-validation.

Theorem 4 shows that the more sophisticated weighted loss estimator achieves exactly the same lower bound as the averaging estimator (Theorem 3). Thus, the weighted loss estimator also does not achieve more than a constant factor improvement over the best of the gold and proxy estimators. Since the weighted estimator spans both the proxy and the gold estimators (depending on the choice of λ), it is to be expected that the best possible weighted estimator does at least as well as either of these two estimators; again, surprisingly, it does no better.

As discussed earlier, prediction error is composed of bias and variance. Training our estimator on the true outcomes alone yields an unbiased but high-variance estimator. On the other hand,

training our estimator on the proxy outcomes alone yields a biased but low-variance estimator. Averaging the estimators or using a weighted loss function can interpolate the bias-variance tradeoff between these two extremes, but provides at most a constant improvement in prediction error.

4. Joint Estimator

We now define our proposed joint estimator, and prove that it can leverage sparsity to achieve much better theoretical guarantees than common approaches used in practice.

4.1. Definition

We propose the following two-step joint estimator $\hat{\beta}_{joint}(\lambda)$:

$$\begin{aligned} \textbf{Step 1:} \quad & \hat{\beta}_{proxy} = \arg\min_{\beta} \left\{ \frac{1}{n_{proxy}} \|Y_{proxy} - \mathbf{X}_{proxy}\beta\|_2^2 \right\} \\ \textbf{Step 2:} \quad & \hat{\beta}_{joint}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{n_{gold}} \|Y_{gold} - \mathbf{X}_{gold}\beta\|_2^2 + \lambda \left\| \beta - \hat{\beta}_{proxy} \right\|_1 \right\}. \end{aligned} \tag{1}$$

Both estimation steps are convex in β. Thus, there are no local minima, and we can find the global minimum through standard techniques such as stochastic gradient descent. Note that the first step only requires proxy data, while the second step only requires gold data; thus, we do not need both gold and proxy data to be simultaneously available during training. This is useful when data from multiple sources cannot be easily combined, but summary information like $\hat{\beta}_{proxy}$ can be shared.

When the regularization parameter λ is small, we recover the gold OLS estimator; when λ is large, we recover the proxy OLS estimator. Thus, similar to model averaging and weighted loss functions, the joint estimator spans both the proxy and the gold estimators (depending on the choice of λ). However, we show that the joint estimator can successfully interpolate the bias-variance tradeoff between these extremes to produce up to an exponential reduction in estimation error.

Intuitively, we seek to do better by leveraging our insight that the bias term δ^* is well-modeled by a sparse function of the covariates. Thus, in principle, we can efficiently recover δ^* using an ℓ_1 penalty. A simple variable transformation of the second-stage objective (1) gives us

$$\hat{\delta}(\lambda) = \arg\min_{\delta} \left\{ \frac{1}{n_{gold}} \left\| Y_{gold} - \mathbf{X}_{gold}(\delta + \hat{\beta}_{proxy}) \right\|_2^2 + \lambda \|\delta\|_1 \right\}, \tag{2}$$

where we have taken $\delta = \beta - \hat{\beta}_{proxy}$. Our estimator is then simply $\hat{\beta}_{joint}(\lambda) = \hat{\delta}(\lambda) + \hat{\beta}_{proxy}$, where $\hat{\beta}_{proxy}$ is estimated in the first stage. In other words, (2) uses the LASSO estimator on gold data to recover the bias term with respect to the proxy estimator $\hat{\beta}_{proxy}$. We use the ℓ_1 penalty, which is known to be effective at recovering sparse vectors (Candes and Tao 2007).

This logic immediately indicates a problem, because the parameter we wish to converge to in (2) is not actually the sparse vector δ^*, but a combination of δ^* and residual noise from the first stage. We formalize this by defining some additional notation:

$$\nu \;=\; \hat{\beta}_{proxy} - \beta^*_{proxy}\,, \tag{3}$$

$$\tilde{\delta} \;=\; \beta^*_{gold} - \hat{\beta}_{proxy} \;=\; \delta^* - \nu\,. \tag{4}$$

Here, ν is the residual noise in estimating the proxy estimator $\hat{\beta}_{proxy}$ from the first stage. As a consequence of this noise, in order to recover the true gold parameter $\beta^*_{gold} = \tilde{\delta} + \hat{\beta}_{proxy}$, we wish to recover $\tilde{\delta}$ (rather than δ^*) from (2). Specifically, note that the minimizer of the first term in (2) is $\tilde{\delta}$ and not δ^*. However, $\tilde{\delta}$ is clearly not sparse, since ν is not sparse (e.g., if the noise ε_{proxy} is a gaussian random variable, then ν is also gaussian). Thus, we may be concerned that the LASSO penalty in (2) may not be able to recover $\tilde{\delta}$ at the exponentially improved rate promised for sparse vectors (Candes and Tao 2007, Bickel et al. 2009, Bühlmann and Van De Geer 2011).

On the other hand, since we have many proxy outcomes (n_{proxy} is large), our proxy estimation error $\|\nu\|_1$ is small. In other words, $\tilde{\delta}$ is *approximately* sparse. We will prove that this is sufficient for us to recover $\tilde{\delta}$ (and therefore β^*_{gold}) at an exponentially improved rate.

4.2. Main Result

We now state a tail inequality that upper bounds the parameter estimation error of the two-step joint estimator with high probability.

THEOREM 5 **(Joint Estimator)**. *The joint estimator satisfies the following tail inequality for any chosen value of the regularization parameter $\lambda > 0$:*

$$\Pr\left[\left\|\hat{\beta}_{joint}(\lambda) - \beta^*_{gold}\right\|_1 \;\geq\; 5\lambda\left(\frac{1}{4\psi^2} + \frac{1}{\psi} + \frac{s}{4\phi^2}\right)\right] \;\leq\; 2d\exp\left(-\frac{\lambda^2 n_{gold}}{200\sigma^2_{gold}}\right) + 2d\exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma^2_{proxy}}\right).$$

The proof is given in Section 4.4 with supporting lemmas in Appendix B. The regularization parameter trades off the bound on the parameter estimation error $\left\|\hat{\beta}_{joint}(\lambda) - \beta^*_{gold}\right\|_1$ with the probability that the bound holds. If λ is too small, the guarantee in Theorem 5 becomes trivial, since the probability of deviation is upper bounded by 1. Thus, λ must be chosen appropriately to achieve a reasonable bound on the parameter estimation error with relatively high probability.

In a typical LASSO problem, an optimal choice of the regularization parameter is $\lambda = \tilde{\mathcal{O}}\left(\sigma_{gold}/\sqrt{n_{gold}}\right)$. However, in Theorem 5, convergence depends on both gold *and* proxy data. In Corollary 1, we will show that in this setting, we will need to choose

$$\lambda = \tilde{\mathcal{O}}\left(\frac{\sigma_{gold}}{\sqrt{n_{gold}}} + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\right).$$

In the next subsection, we will compute the resulting estimation error of the joint estimator.

4.3. Comparison with Baselines

We now derive an upper bound on the expected parameter estimation error of the joint estimator, in order to compare its performance against the baseline estimators described in Section 3.

From Theorem 5, we know that our estimation error $\left\|\hat{\beta}_{joint} - \beta^*_{gold}\right\|_1$ is small with high probability. However, to derive an upper bound on $R(\cdot)$, we also need to characterize its worst-case magnitude. In order to ensure that our estimator $\hat{\beta}_{joint}$ never becomes unbounded, we consider the *truncated* joint estimator $\hat{\beta}^{tr}_{joint}$. In particular,

$$\hat{\beta}^{tr}_{joint} = \begin{cases} \hat{\beta}_{joint} & \text{if } \left\|\hat{\beta}_{joint}\right\|_1 \leq 2b\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Recall that b is any upper bound on $\left\|\beta^*_{gold}\right\|_1$ (Assumption 1), and can simply be considered a large constant. The following corollary uses the tail inequality in Theorem 5 to obtain an upper bound on the expected parameter estimation error of the truncated joint estimator.

COROLLARY 1 **(Joint Estimator)**. *The parameter estimation error of the truncated joint estimator on both gold and proxy data* $(\mathbf{X}_{gold}, Y_{gold}, \mathbf{X}_{proxy}, Y_{proxy})$ *is bounded above as follows:*

$$R\left(\hat{\beta}^{tr}_{joint}(\lambda), \beta^*_{gold}\right) \;\leq\; 5\lambda\left(\frac{1}{4\psi^2} + \frac{1}{\psi} + \frac{s}{4\phi^2}\right) + 6bd\left(\exp\left(-\frac{\lambda^2 n_{gold}}{200\sigma^2_{gold}}\right) + \exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma^2_{proxy}}\right)\right).$$

Let $C > 0$ *be any tuning constant. Taking the regularization parameter to be*

$$\bar{\lambda} \;=\; C\max\left\{\sqrt{\frac{200\sigma^2_{gold}\log\left(6bdn_{gold}\right)}{n_{gold}}},\; \sqrt{\frac{2d^2\sigma^2_{proxy}\log\left(6bdn_{proxy}\right)}{n_{proxy}}}\right\} = \;\tilde{\mathcal{O}}\left(\frac{\sigma_{gold}}{\sqrt{n_{gold}}} + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\right),$$

yields a parameter estimation error of order

$$R\left(\hat{\beta}^{tr}_{joint}(\bar{\lambda}), \beta^*_{gold}\right) \;=\; \mathcal{O}\left(\max\left\{\frac{s\sigma_{gold}}{\sqrt{n_{gold}}}\log\left(dn_{gold}\right),\; \frac{sd\sigma_{proxy}}{\sqrt{n_{proxy}}}\log\left(dn_{proxy}\right)\right\}\right).$$

The proof is given in Appendix B.2. The parameter estimation error $R\left(\hat{\beta}^{tr}_{joint}(\bar{\lambda}), \beta^*_{gold}\right)$ cleanly decomposes into two terms: (i) the first term is the classical error rate of the LASSO estimator *if* β^*_{gold} (rather than δ^*) was sparse, and (ii) the second term is proportional to the error of the proxy estimator *if* there were no bias (i.e., $\delta^* = 0$).

Comparison: For ease of comparison, we tabulate the bounds we have derived so far (up to constants and logarithmic factors) in Table 1. Recall that we are interested in the regime where n_{proxy} is large and n_{gold} is small. Even with infinite proxy samples, the proxy estimator's error is bounded below by its bias $\|\delta^*\|_1$. The gold estimator's error can also be very large, particularly when $n_{gold} \lesssim d$. Model averaging and weighted loss functions do not improve this picture by more than a constant factor. Now, note that in our regime of interest,

$$\frac{s\sigma_{gold}}{\sqrt{n_{gold}}} \;\ll\; \frac{d\sigma_{gold}}{\sqrt{n_{gold}}} \quad \text{and} \quad \frac{sd\sigma_{proxy}}{\sqrt{n_{proxy}}} \;\ll\; \|\delta^*\|_1 + \frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\,.$$

The first claim follows when $s \ll d$ (i.e., the bias term δ^* is reasonably sparse), and the second claim follows when $\|\delta^*\|_1 \gg sd\sigma_{proxy}/\sqrt{n_{proxy}}$ (i.e., the proxy estimator's error primarily arises from its

Estimator	**Parameter Estimation Error** *(up to constants)*	**Bound Type**
Gold OLS	$\frac{d\sigma_{gold}}{\sqrt{n_{gold}}}$	Lower
Gold Ridge	$\frac{d\sigma_{gold}}{\sqrt{n_{gold}}+d\sigma_{gold}}$	Lower
Proxy OLS	$\|\delta^*\|_1+\frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}$	Lower
Averaging	$\min\left\{\frac{d\sigma_{gold}}{\sqrt{n_{gold}}},\ \|\delta^*\|_1+\frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\right\}$	Lower
Weighted	$\min\left\{\frac{d\sigma_{gold}}{\sqrt{n_{gold}}},\ \|\delta^*\|_1+\frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\right\}$	Lower
Truncated Joint	$\max\left\{\frac{s\sigma_{gold}}{\sqrt{n_{gold}}}\log\left(dn_{gold}\right),\ \frac{sd\sigma_{proxy}}{\sqrt{n_{proxy}}}\log\left(dn_{proxy}\right)\right\}$	Upper

Table 1 Comparison of parameter estimation error across estimators.

bias δ^* rather than its variance, and s is small). Thus, the joint estimator's error can be significantly lower than popular heuristics in our regime of interest.

Sample Complexity: We can also interpret these results in terms of sample complexity. Let the decision-maker target a nontrivial parameter estimation error $\xi < \|\delta^*\|_1$. As noted earlier, even an infinite number of proxy observations will not suffice, and one requires some gold data. Based on the bounds in Table 1, it can easily be verified that the gold OLS/ridge, model averaging and weighted loss functions require $n_{gold} = \mathcal{O}\left(d^2\sigma_{gold}^2/\xi^2\right)$ *regardless* of the number of proxy observations. In contrast, the joint estimator only requires $n_{gold} = \mathcal{O}\left(s^2\sigma_{gold}^2\log^2\left(d\cdot\frac{s\sigma_{gold}}{\xi}\right)/\xi^2\right)$ as long as $n_{proxy} \gtrsim \mathcal{O}\left(s^2d^2\sigma_{proxy}^2\log^2\left(d\cdot\frac{s\sigma_{proxy}}{\xi}\right)/\xi^2\right)$. In other words, when sufficient proxy data is available, the number of gold observations required is exponentially smaller in the dimension d.

4.4. Proof of Theorem 5

We start by defining the following two events:

$$\mathcal{J} = \left\{\frac{2}{n_{gold}}\left\|\varepsilon_{gold}^\top \mathbf{X}_{gold}\right\|_\infty \le \lambda_0\right\}, \tag{5}$$

$$\mathcal{I} = \left\{\left\|\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2^2 \le \lambda_1\right\}, \tag{6}$$

where we have introduced two new parameters λ_0 and λ_1. We denote the complements of these events as $\mathcal{J}^C$ and $\mathcal{I}^C$ respectively. When events $\mathcal{J}$ and $\mathcal{I}$ hold, the gold and proxy noise terms ε_{gold} and ε_{proxy} are bounded in magnitude, allowing us to bound our parameter estimation error $\left\|\tilde{\delta}-\hat{\delta}\right\|_1$. Since our noise is subgaussian, $\mathcal{J}$ and $\mathcal{I}$ hold with high probability (Lemmas 4 and 5). We will choose the parameters λ_0 and λ_1 later to optimize our bounds.

LEMMA 1. *On the event $\mathcal{J}$, taking $\lambda \geq 5\lambda_0$, the solution $\hat{\delta}$ to the optimization problem (2) satisfies*

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;\leq\; \frac{5}{4n_{gold}} \left\| \mathbf{X}_{gold} \nu \right\|_2^2 + 5\lambda \left\| \nu \right\|_1 + \frac{5\lambda^2 s}{4\phi^2} .$$

Proof of Lemma 1 Since the optimization problem (2) is convex, it recovers the in-sample global minimum. Thus, we must have that

$$\frac{1}{n_{gold}} \left\| Y_{gold} - \mathbf{X}_{gold} \left(\hat{\delta} + \hat{\beta}_{proxy} \right) \right\|_2^2 + \lambda \left\| \hat{\delta} \right\|_1 \;\leq\; \frac{1}{n_{gold}} \left\| Y_{gold} - \mathbf{X}_{gold} \left(\tilde{\delta} + \hat{\beta}_{proxy} \right) \right\|_2^2 + \lambda \left\| \tilde{\delta} \right\|_1 .$$

Substituting $Y_{gold} = \mathbf{X}_{gold}\beta^*_{gold} + \varepsilon_{gold} = \mathbf{X}_{gold} \left(\tilde{\delta} + \hat{\beta}_{proxy} \right) + \varepsilon_{gold}$ yields

$$\frac{1}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) + \varepsilon_{gold} \right\|_2^2 + \lambda \left\| \hat{\delta} \right\|_1 \;\leq\; \frac{1}{n_{gold}} \left\| \varepsilon_{gold} \right\|_2^2 + \lambda \left\| \tilde{\delta} \right\|_1 .$$

Expanding $\left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) + \varepsilon_{gold} \right\|_2^2 = \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + \left\| \varepsilon_{gold} \right\|_2^2 + 2\varepsilon_{gold}^\top \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right)$ and cancelling terms on both sides gives us

$$\frac{1}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + \lambda \left\| \hat{\delta} \right\|_1 \;\leq\; \frac{2}{n_{gold}} \varepsilon_{gold}^\top \mathbf{X}_{gold} \left(\hat{\delta} - \tilde{\delta} \right) + \lambda \left\| \tilde{\delta} \right\|_1 . \tag{7}$$

By Hölder's inequality, when $\mathcal{J}$ holds and $\lambda \geq 5\lambda_0$, we have

$$\begin{aligned} \frac{2}{n_{gold}} \varepsilon_{gold}^\top \mathbf{X}_{gold} \left(\hat{\delta} - \tilde{\delta} \right) \;&\leq\; \frac{2}{n_{gold}} \left\| \varepsilon_{gold}^\top \mathbf{X}_{gold} \right\|_\infty \cdot \left\| \hat{\delta} - \tilde{\delta} \right\|_1 \\ &\leq\; \frac{\lambda}{5} \left\| \hat{\delta} - \tilde{\delta} \right\|_1 . \end{aligned}$$

Substituting into Eq. (7), we have on $\mathcal{J}$ that

$$\begin{aligned} \frac{5}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + 5\lambda \left\| \hat{\delta} \right\|_1 \;&\leq\; \lambda \left\| \hat{\delta} - \tilde{\delta} \right\|_1 + 5\lambda \left\| \tilde{\delta} \right\|_1 \\ &=\; \lambda \left\| \hat{\delta} - \delta^* + \nu \right\|_1 + 5\lambda \left\| \delta^* - \nu \right\|_1 , \end{aligned} \tag{8}$$

where we recall that $\nu = \delta^* - \tilde{\delta}$. The second line uses ν to express the right hand side in terms of δ^* so that we can ultimately invoke the compatibility condition on $\hat{\delta}$ (Definition 2). To do this, we must first express $\hat{\delta}$ in terms of its components on the index set $S = supp(\delta^*)$.

By the triangle inequality, we have

$$\begin{aligned} \left\| \hat{\delta} \right\|_1 \;&=\; \left\| \hat{\delta}_S \right\|_1 + \left\| \hat{\delta}_{S^c} \right\|_1 \\ &\geq\; \left\| \delta^*_S \right\|_1 - \left\| \hat{\delta}_S - \delta^*_S \right\|_1 + \left\| \hat{\delta}_{S^c} \right\|_1 . \end{aligned} \tag{9}$$

Similarly, noting that $\delta^*_{S^c} = 0$ by definition of S, we have

$$\left\| \hat{\delta} - \delta^* + \nu \right\|_1 \;\leq\; \left\| \hat{\delta}_S - \delta^*_S \right\|_1 + \left\| \hat{\delta}_{S^c} \right\|_1 + \left\| \nu \right\|_1 . \tag{10}$$

Collecting Eq. (9)–(10) and substituting into Eq. (8), we have that when $\mathcal{J}$ holds,

$$\frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 \;\leq\; 6\lambda\left\|\hat{\delta}_S-\delta^*_S\right\|_1+6\lambda\left\|\nu\right\|_1. \tag{11}$$

Ideally, we would now invoke the compatibility condition (Definition 2) to $u=\hat{\delta}-\delta^*$ to bound $\left\|\hat{\delta}_S-\delta^*_S\right\|_1$ in Eq. (11) above. However, this requires u to satisfy $\|u_{S^c}\|_1\leq 3\|u_S\|_1$, which may not hold in general. Thus, we proceed by considering two cases: either (i) $\|\nu\|_1\leq\left\|\hat{\delta}_S-\delta^*_S\right\|_1$, or (ii) $\left\|\hat{\delta}_S-\delta^*_S\right\|_1<\|\nu\|_1$. In Case (i), we will invoke the compatibility condition to prove our finite-sample guarantee for the joint estimator, and in Case (ii), we will find that we already have good control over the error of the estimator.

Case (i): We are in the case that $\|\nu\|_1\leq\left\|\hat{\delta}_S-\delta^*_S\right\|_1$, so from Eq. (11), we can write on $\mathcal{J}$,

$$\frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 \;\leq\; 12\lambda\left\|\hat{\delta}_S-\delta^*_S\right\|_1.$$

Dropping the first (non-negative) term on the left hand side, we immediately observe that

$$\left\|\hat{\delta}_{S^c}\right\|_1=\left\|\hat{\delta}_{S^c}-\delta^*_{S^c}\right\|_1 \;\leq\; 3\left\|\hat{\delta}_S-\delta^*_S\right\|_1,$$

so we can apply the compatibility condition to $u=\hat{\delta}-\delta^*$. This yields

$$\left\|\hat{\delta}_S-\delta^*_S\right\|_1^2 \;\leq\; \frac{s}{\phi^2}\left(\hat{\delta}-\delta^*\right)\Sigma_{gold}\left(\hat{\delta}-\delta^*\right).$$

Taking the square-root, we get

$$\left\|\hat{\delta}_S-\delta^*_S\right\|_1 \;\leq\; \frac{\sqrt{s}}{\phi\sqrt{n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta}-\delta^*\right)\right\|_2. \tag{12}$$

Separately, when Case (i) and $\mathcal{J}$ hold, we can further simplify

$$\begin{aligned}
\frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\tilde{\delta}-\hat{\delta}\right\|_1 \;&=\; \frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\hat{\delta}-\delta^*+\nu\right\|_1\\
&\leq\; \frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\hat{\delta}_S-\delta^*_S\right\|_1+4\lambda\left\|\hat{\delta}_{S^c}\right\|_1\\
&\quad+4\lambda\left\|\nu\right\|_1\\
&\leq\; 10\lambda\left\|\hat{\delta}_S-\delta^*\right\|_1+10\lambda\left\|\nu\right\|_1,
\end{aligned}$$

where we used Eq. (10) in the first inequality and Eq. (11) in the second inequality. We can now proceed by applying Eq. (12)

$$\begin{aligned}
\frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+4\lambda\left\|\tilde{\delta}-\hat{\delta}\right\|_1 \;&\leq\; \frac{10\lambda\sqrt{s}}{\phi\sqrt{n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta}-\delta^*\right)\right\|_2+10\lambda\left\|\nu\right\|_1\\
&\leq\; \frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\hat{\delta}-\delta^*\right)\right\|_2^2+10\lambda\left\|\nu\right\|_1+\frac{5\lambda^2 s}{4\phi^2}\\
&\leq\; \frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\tilde{\delta}-\hat{\delta}\right)\right\|_2^2+\frac{5}{n_{gold}}\left\|\mathbf{X}_{gold}\nu\right\|_2^2+10\lambda\left\|\nu\right\|_1\\
&\quad+\frac{5\lambda^2 s}{4\phi^2},
\end{aligned}$$

where the second inequality follows from the fact that $10ab \leq 5a^2 + 5b^2$ for any $a, b \in \mathbb{R}$. Then, when $\mathcal{J}$ and Case (i) hold, we have that

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;\leq\; \frac{5}{4n_{gold}} \left\| \mathbf{X}_{gold} \nu \right\|_2^2 + \frac{5\lambda}{2} \left\| \nu \right\|_1 + \frac{5\lambda^2 s}{4\phi^2} \,. \tag{13}$$

Case (ii): We are in the case that $\left\| \hat{\delta}_S - \delta_S^* \right\|_1 \leq \left\| \nu \right\|_1$, so Eq. (11) implies on $\mathcal{J}$,

$$\frac{5}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + 4\lambda \left\| \hat{\delta}_{S^c} \right\|_1 \;\leq\; 12\lambda \left\| \nu \right\|_1 \,. \tag{14}$$

In this case, we do not actually need to invoke the compatibility condition. When $\mathcal{J}$ and Case (ii) hold, we can directly bound

$$\begin{aligned} \frac{5}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + 4\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;&\leq\; \frac{5}{n_{gold}} \left\| \mathbf{X}_{gold} \left(\tilde{\delta} - \hat{\delta} \right) \right\|_2^2 + 4\lambda \left\| \hat{\delta}_S - \delta_S^* \right\|_1 + 4\lambda \left\| \hat{\delta}_{S^c} \right\|_1 \\ &\quad + 4\lambda \left\| \nu \right\|_1 \\ &\leq\; 20\lambda \left\| \nu \right\|_1 \,, \end{aligned}$$

where we used Eq. (10) in the first inequality and Eq. (14) as well as the fact that $\left\| \hat{\delta}_S - \delta_S^* \right\|_1 \leq \left\| \nu \right\|_1$ in the second inequality. Dropping the first (non-negative) term on the left hand side yields

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;\leq\; 5\lambda \left\| \nu \right\|_1 \,. \tag{15}$$

Combining the inequalities from Eq. (13) and (15), the following holds in both cases on $\mathcal{J}$,

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;\leq\; \frac{5}{4n_{gold}} \left\| \mathbf{X}_{gold} \nu \right\|_2^2 + 5\lambda \left\| \nu \right\|_1 + \frac{5\lambda^2 s}{4\phi^2} \,. \tag{16}$$

□

In other words, we have shown that we can bound $\left\| \tilde{\delta} - \hat{\delta} \right\|_1$ with high probability when ν (the approximation error of $\hat{\beta}_{proxy}$) is small. We expect this error to be small since the number of proxy samples n_{proxy} is large. The next lemma bounds the terms that depend on ν on the event $\mathcal{I}$. The proof is given in Appendix B.1.

LEMMA 2. *On the event $\mathcal{I}$, we have that both*

$$\left\| \mathbf{X}_{gold} \nu \right\|_2^2 \leq \frac{d n_{gold}}{\psi^2 n_{proxy}^2} \lambda_1 \,, \qquad \textit{and} \qquad \left\| \nu \right\|_1 \leq \frac{\sqrt{d\lambda_1}}{\psi n_{proxy}} \,.$$

The next lemma simply applies these bounds on ν to the bound we derived earlier on $\left\| \tilde{\delta} - \hat{\delta} \right\|_1$.

LEMMA 3. *On the events $\mathcal{J}$ and $\mathcal{I}$, taking $\lambda \geq 5\lambda_0$, the solution $\hat{\delta}$ to the optimization problem (2) satisfies*

$$\left\| \tilde{\delta} - \hat{\delta} \right\|_1 \;\leq\; \frac{5d\lambda_1}{4\psi^2 n_{proxy}^2 \lambda} + \frac{5\sqrt{d\lambda_1}}{\psi n_{proxy}} + \frac{5\lambda s}{4\phi^2} \,.$$

Proof of Lemma 3 The bound follows from applying Lemma 2 to the result in Lemma 1. □

Lemma 3 shows that we can bound our parameter estimation error on the events $\mathcal{J}$ and $\mathcal{I}$. The next two lemmas use a concentration inequality for subgaussian random variables to show that these events hold with high probability. Their proofs are given in Appendix B.1.

LEMMA 4. *The probability of event $\mathcal{J}$ is bounded by*

$$\Pr[\mathcal{J}] \;\geq\; 1 - 2d\exp\left(-\frac{\lambda_0^2 n_{gold}}{8\sigma_{gold}^2}\right).$$

LEMMA 5. *The probability of event $\mathcal{I}$ is bounded by*

$$\Pr[\mathcal{I}] \;\geq\; 1 - 2d\exp\left(-\frac{\lambda_1}{2d\sigma_{proxy}^2 n_{proxy}}\right).$$

We now combine Lemmas 3, 4, and 5, and choose values of our parameters λ_0 and λ_1 to complete our proof of Theorem 5.

Proof of Theorem 5 By Lemma 3, the following holds with probability 1 when the events $\mathcal{J}$ and $\mathcal{I}$ hold, and $\lambda \geq 5\lambda_0$

$$\left\|\tilde{\delta} - \hat{\delta}\right\|_1 \;\leq\; \frac{5d\lambda_1}{4\psi^2 n_{proxy}^2 \lambda} + \frac{5\sqrt{d\lambda_1}}{\psi n_{proxy}} + \frac{5\lambda s}{4\phi^2}.$$

Recall that λ_0, λ_1 are theoretical quantities that we can choose freely to optimize our bound. In contrast, λ is a fixed regularization parameter chosen by the decision-maker when training the estimator. Then, setting $\lambda_0 = \lambda/5$, we can write

$$\begin{aligned}
\Pr\left[\left\|\tilde{\delta} - \hat{\delta}\right\|_1 \;\geq\; \frac{5d\lambda_1}{4\psi^2 n_{proxy}^2 \lambda} + \frac{5\sqrt{d\lambda_1}}{\psi n_{proxy}} + \frac{5\lambda s}{4\phi^2}\right] \;&\leq\; 1 - \Pr[\mathcal{J} \cap \mathcal{I}] \\
&\leq\; \Pr[\mathcal{J}^C] + \Pr[\mathcal{I}^C] \\
&\leq\; 2d\exp\left(-\frac{\lambda^2 n_{gold}}{200\sigma_{gold}^2}\right) + 2d\exp\left(-\frac{\lambda_1}{2d\sigma_{proxy}^2 n_{proxy}}\right).
\end{aligned}$$

The second inequality follows from a union bound, and the third follows from Lemma 4 (setting $\lambda_0 = \lambda/5$) and Lemma 5. By inspection, we choose

$$\lambda_1 = \frac{n_{proxy}^2 \lambda^2}{d},$$

yielding

$$\Pr\left[\left\|\tilde{\delta} - \hat{\delta}\right\|_1 \;\geq\; 5\lambda\left(\frac{1}{4\psi^2} + \frac{1}{\psi} + \frac{s}{4\phi^2}\right)\right] \;\leq\; 2d\exp\left(-\frac{\lambda^2 n_{gold}}{200\sigma_{gold}^2}\right) + 2d\exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma_{proxy}^2}\right).$$

Finally, we reverse our variable transformation by substituting $\hat{\beta}_{joint} = \hat{\delta} + \hat{\beta}_{proxy}$ and $\beta_{gold}^* = \tilde{\delta} + \hat{\beta}_{proxy}$, which gives us the result. □

4.5. Nonlinear Predictors

Thus far, we have focused on linear predictors. Our joint estimator in Section 4.1 can be adapted to any M-estimator given its parametric empirical loss function $\ell(\cdot)$ as follows:

$$\textbf{Step 1:} \quad \hat{\beta}_{proxy} = \arg\min_{\beta} \left\{ \frac{1}{n_{proxy}} \sum_{i=1}^{n_{proxy}} \ell\left(\beta;\ \mathbf{X}^{(i)}_{proxy}, Y^{(i)}_{proxy}\right) \right\},$$

$$\textbf{Step 2:} \quad \hat{\beta}_{joint}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \ell\left(\beta;\ \mathbf{X}^{(i)}_{gold}, Y^{(i)}_{gold}\right) + \lambda \left\| \beta - \hat{\beta}_{proxy} \right\|_1 \right\}. \tag{17}$$

Our proof techniques for bounding the resulting parameter estimation error generalize straightforwardly as long as $\ell(\cdot)$ is convex and satisfies mild technical assumptions (in particular, the classical margin condition, e.g., see Negahban et al. 2009, Bühlmann and Van De Geer 2011). To illustrate, in this section, we extend our theoretical guarantees to the family of generalized linear models.

Preliminaries: We start by defining some additional notation. Under the variable transformation $\delta = \beta - \hat{\beta}_{proxy}$, let the in-sample loss function on the training set $(\mathbf{X}, \mathbf{Y})$ be

$$\mathcal{L}(\delta;\ \mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(\delta + \hat{\beta}_{proxy};\ \mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\right).$$

Then, the second-stage objective (17) can be re-written as

$$\hat{\delta}(\lambda) = \arg\min_{\delta}\ \mathcal{L}(\delta;\ \mathbf{X}_{gold}, Y_{gold}) + \lambda \left\| \delta \right\|_1. \tag{18}$$

Once again, we define the optimal parameter $\tilde{\delta} = \min_{\delta} \mathbb{E}_{\varepsilon_{gold}} \mathcal{L}(\delta)$ (recall that $\tilde{\delta} \neq \delta^*$ because we only have access to $\hat{\beta}_{proxy}$ rather than β^*_{proxy}). We define the empirical process $w(\delta)$, and the expected error $\mathcal{E}(\delta)$ relative to the expected error of the optimal parameter $\tilde{\delta}$ respectively as

$$w(\delta) = \mathcal{L}(\delta) - \mathbb{E}_{\varepsilon_{gold}} \mathcal{L}(\delta),$$
$$\mathcal{E}(\delta) = \mathbb{E}_{\varepsilon_{gold}} \mathcal{L}(\delta) - \mathbb{E}_{\varepsilon_{gold}} \mathcal{L}(\tilde{\delta}).$$

The margin condition essentially states that $\mathcal{E}(\delta)$ is lower bounded by a positive quantity that scales with $\left\| \delta - \tilde{\delta} \right\|$, i.e., the loss function $\ell(\cdot)$ evaluated on $(\mathbf{X}_{gold}, Y_{gold})$ cannot be "flat" around the optimal parameter $\tilde{\delta}$; if this is not the case, then we cannot distinguish the optimal parameter from nearby parameters on the training data. This is a standard assumption in the classification literature (Tsybakov et al. 2004), and has previously been adapted to studying nonlinear M-estimators in high dimension (Negahban et al. 2009). Generalized linear models with strongly convex inverse link functions naturally satisfy a quadratic margin condition (see Lemma 6).

Generalized Linear Models: A generalized linear model with parameter β and observation $\mathbf{x}$ has outcomes distributed as

$$y \sim \exp\left(y \mathbf{x}^\top \beta - A\left(\mathbf{x}^\top \beta\right) + B\left(y\right) \right),$$

where A and B are known functions. Under this model, $\mathbb{E}[y \mid \mathbf{x}] = \mu(\mathbf{x}^\top \beta)$ and $\mathrm{Var}[y \mid \mathbf{x}] = \mu'(\mathbf{x}^\top \beta)$, where $\mu = A'$ is the *inverse link function*. For instance, in logistic regression, we have binary outcomes y with $\mu(z) = 1/(1+\exp(-z))$; in Poisson regression, we have integer-valued outcomes y with $\mu(z) = \exp(z)$; in linear regression, we have continuous outcomes y with $\mu(z) = z$.

The resulting maximum likelihood estimator is $\hat{\beta} = \arg\min_\beta \sum_{i=1}^n \left\{-y_i \mathbf{x}_i^\top \beta + A(\mathbf{x}_i^\top \beta) - B(y_i)\right\}$ (see, e.g., McCullagh and Nelder 1989), implying the corresponding in-sample loss function

$$\mathcal{L}(\delta;\ \mathbf{X}, \mathbf{Y}) = \sum_{i=1}^n \left\{-y_i \mathbf{x}_i^\top \left(\delta + \hat{\beta}_{proxy}\right) + A\left(\mathbf{x}_i^\top \left(\delta + \hat{\beta}_{proxy}\right)\right) - B(y_i)\right\}. \tag{19}$$

In order to ensure convexity and the margin condition, we now impose that A is strongly convex. This assumption is mild and is satisfied by any strictly increasing inverse link function, e.g., all the examples given above (logistic, Poisson, linear) satisfy strong convexity on a bounded domain. Note that Assumption 1 and our deterministic design matrix ensure that our domain is bounded.

Assumption 4. *The function $A(\cdot)$ is strongly convex with parameter $m > 0$, i.e., for all points x, x' in its domain,*

$$A(x') - A(x) \ \geq\ A'(x) \cdot (x' - x) + m(x' - x)^2/2\,.$$

As noted earlier, generalized linear models with strongly convex inverse link functions naturally satisfy a quadratic margin condition as follows.

Lemma 6 **(Margin Condition)**. *For any δ in its domain, a generalized linear model satisfying Assumption 4 also satisfies*

$$\mathcal{E}(\delta) \ \geq\ \frac{m}{2n_{gold}} \left\|\mathbf{X}_{gold}\left(\delta - \tilde{\delta}\right)\right\|_2^2.$$

The proof of Lemma 6 is given in Appendix C.1. This result ensures that we can distinguish $\tilde{\delta}$ from other parameters on the training data when there is no noise.

We now have all the pieces required to establish a tail inequality that upper bounds the parameter estimation error of the two-step joint estimator with high probability for generalized linear models.

Theorem 6. *The nonlinear joint estimator satisfies the following tail inequality for a generalized linear model and any chosen value of the regularization parameter $\lambda > 0$:*

$$\Pr\left[\left\|\hat{\beta}_{joint}(\lambda) - \beta^*_{gold}\right\|_1 \ \geq\ \frac{5\lambda}{m}\left(\frac{1}{8\psi^2} + \frac{1}{\psi} + \frac{s}{2\phi^2}\right)\right] \ \leq\ 2d\exp\left(-\frac{\lambda^2 n_{gold}}{50\sigma^2_{gold}}\right) + 2d\exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma^2_{proxy}}\right).$$

The proof of Theorem 6 and associated lemmas is given in Appendix C. Note that this result is nearly identical to the result in the linear case (Theorem 5), with some small changes to the constants and an additional dependence on the strong convexity parameter m related to the inverse link function. As a result, the expected parameter estimation error also satisfies the same bound as the linear case up to constants (see Corollary 2 in Appendix C.3).

4.6. Remarks

We now briefly discuss applying our proposed estimator in practice.

Cross-validation: While Corollary 1 specifies a theoretically good choice for the regularization parameter λ, this choice depends on problem-specific parameters that are typically unknown. In practice, λ is typically chosen using the popular heuristic of cross-validation (Picard and Cook 1984, Friedman et al. 2001). In Appendix E.1, we show that the two approaches attain very similar performance. A related literature investigates the consistency of estimators that use cross-validation for model selection (see, e.g., Li et al. 1987, Homrighausen and McDonald 2014); future work could analogously study asymptotic properties of the joint estimator with cross-validation.

Scaling: When the gold and proxy outcomes are different, it may be useful to perform a pre-processing step to ensure that both outcomes have similar magnitude. In Example 1, clicks are roughly $10\times$ as frequent as purchases. Thus, in our numerical experiments, we scale down the responses Y_{proxy} by this factor to ensure that the responses are of similar magnitude in expectation. i.e., $\mathbb{E}[|y_{gold}(\mathbf{x})|] \approx \mathbb{E}[|y_{proxy}(\mathbf{x})|]$ for the same feature vector $\mathbf{x}$. The scaling constant is typically known, or can be easily estimated from a hold-out set. While this step is not required for the theory, it helps increase the similarity between β^*_{gold} and β^*_{proxy}, making it more likely that we can successfully estimate the bias δ^* using a simple (sparse) function.

Combined estimation: Our proposed two-step estimator does not require the simultaneous availability of gold and proxy data for training. This is an important feature in settings such as healthcare, where data from different sources often cannot be combined due to regulatory constraints. It also yields a simpler statistical analysis. However, if both sources of data are available together, one could alternatively combine the two-step estimation procedure, and directly estimate both $\hat{\beta}_{gold}(\lambda)$ and $\hat{\beta}_{proxy}(\lambda)$ using the following heuristic:

$$\left\{\hat{\beta}_{gold}(\lambda), \hat{\beta}_{proxy}(\lambda)\right\} = \underset{\beta_{gold},\beta_{proxy}}{\arg\min} \left\{ \underbrace{\|Y_{gold} - \mathbf{X}_{gold}\beta_{gold}\|_2^2}_{\Theta(n_{gold})} + \underbrace{\|Y_{proxy} - \mathbf{X}_{proxy}\beta_{proxy}\|_2^2}_{\Theta(n_{proxy})} + \lambda \|\beta_{proxy} - \beta_{gold}\|_1 \right\}.$$

We suggest choosing the regularization parameter $\lambda = \tilde{\mathcal{O}}\left(\sigma_{gold}\sqrt{n_{gold}} + \frac{d\sigma_{proxy}n_{gold}}{\sqrt{n_{proxy}}}\right)$ to match the normalization suggested by Corollary 1. The expression above is for the linear case, but one can also consider nonlinear generalizations as in Section 4.5.

It is worth examining the regime where $n_{proxy} \gg n_{gold}$. In this case, the combined estimator actually decouples into our two-step procedure in Eq. (1). This is because the second term, which scales as $\Theta(n_{proxy})$, dominates the objective function so $\hat{\beta}_{proxy} \approx \arg\min_{\beta_{proxy}} \|Y_{proxy} - \mathbf{X}_{proxy}\beta_{proxy}\|_2^2$. Once this value is fixed for $\hat{\beta}_{proxy}$, the remaining optimization problem over β_{gold} trivially reduces to the second step of our proposed estimator. Consistent with this observation, we simulate the

combined estimator in Appendix E.1, and find that its performance essentially matches that of the two-step joint estimator. Next, note that the first term of the combined estimator's objective scales as $\Theta(n_{gold})$ while the third term scales as $\Theta(\sqrt{n_{gold}})$ when $n_{proxy} \gg n_{gold}$. Thus, when n_{gold} is small, the regularization term will be large relative to the first term so $\hat{\beta}_{gold}$ will be strongly regularized towards $\hat{\beta}_{proxy}$; however, when n_{gold} becomes large, the first term will dominate the third term so we will eventually obtain the OLS estimate $\hat{\beta}_{gold} \approx \arg\min_{\beta_{gold}} \|Y_{gold} - \mathbf{X}_{gold}\beta_{gold}\|_2^2$.

5. Experiments

We now test the performance of our proposed joint estimator against benchmark estimators on both synthetic and real datasets.

5.1. Synthetic

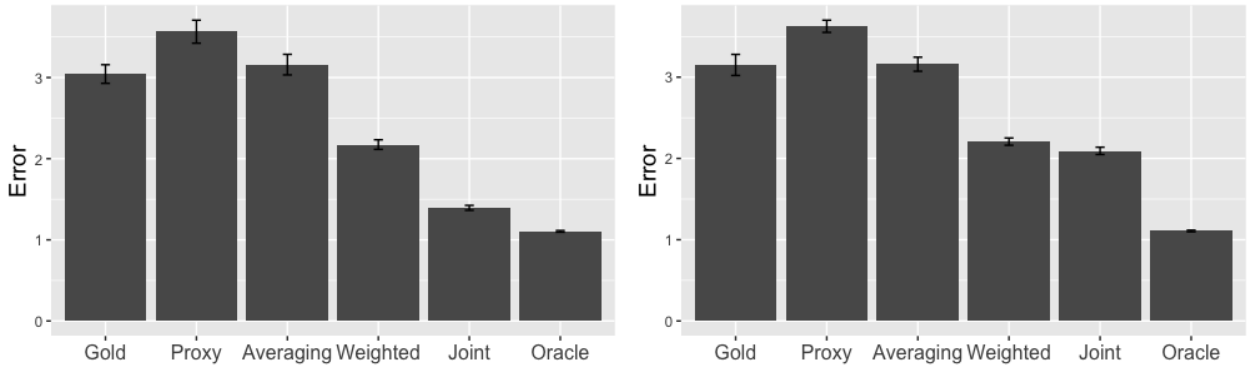
We will consider two cases: (i) a sparse bias term δ^* (matching our assumptions and analysis), and (ii) a non-sparse bias term δ^* (i.e., $s = d$).

Data Generation: We set $n_{proxy} = 1000$, $n_{gold} = 150$, $n_{test} = 1000$, $d = 100$, and fix our true parameter $\beta^*_{gold} = \mathbf{1} \in \mathbb{R}^d$. We generate our proxy observations $\mathbf{X}_{proxy} \in \mathbb{R}^{n_{proxy} \times d}$ from a multivariate normal distribution with mean $\mathbf{0}$ and a random covariance matrix generated as follows: (i) draw a random matrix in $\mathbb{R}^{d \times d}$ whose entries are uniform random samples from $[0, 1]$, (ii) multiply the resulting matrix with its transpose to ensure that it is positive-definite, and (iii) normalize it with its trace. We take $\mathbf{X}_{gold} \in \mathbb{R}^{n_{gold} \times d}$ to simply be the first n_{gold} rows of $\mathbf{X}_{proxy}$, and we additionally generate a test set $\mathbf{X}_{test} \in \mathbb{R}^{n_{test} \times d}$ in the same way we generate $\mathbf{X}_{proxy}$. Our data-generating process is the simple linear model, with $Y_{proxy} = \mathbf{X}_{gold}\left(\beta^*_{gold} - \delta^*\right) + \varepsilon_{proxy}$, $Y_{gold} = \mathbf{X}_{gold}\beta^*_{gold} + \varepsilon_{gold}$, and test set responses $Y_{test} = \mathbf{X}_{test}\beta^*_{gold} + \varepsilon_{test}$. All noise terms are $\varepsilon_{proxy}, \varepsilon_{gold}, \varepsilon_{test} \sim \mathcal{N}(0, 1)$.

We study both sparse and non-sparse realizations of δ^*. In the sparse case, δ^* is a randomly drawn binomial vector $0.1 \times B(d, 0.1)$, i.e., $s \ll d$. In the non-sparse case, δ^* is a randomly drawn gaussian vector $\mathcal{N}(\mathbf{0}, 0.15 \times I_d)$, i.e., $s = d$. These parameters were chosen to keep the performance of the proxy and gold estimators relatively similar in both cases.

Estimators: The gold and proxy estimators are simply OLS estimators on gold and proxy data respectively. The averaging, weighted, and joint estimators require a tuning parameter λ: for these, we split the gold observations randomly, taking 70% to be the training set and the remaining 30% to be the validation set. We then train models with different values of λ on the training set, and use the mean squared error on the validation set to choose the best value of λ for each estimator in the final model (Friedman et al. 2001). Finally, we consider an "Oracle" benchmark that has advance knowledge of the true (random) bias term δ^*, and adjusts the proxy estimator accordingly.

Evaluation: Our evaluation metric is the average out-of-sample prediction error $\frac{1}{n_{test}} \left\| Y_{test} - \hat{Y} \right\|_2^2$, where $\hat{Y}$ are the predictions of an estimator on the test set $\mathbf{X}_{test}$. We average our results over 100 trials, where we randomly draw all problem parameters in each iteration.



(a) Sparse bias term

(b) Non-sparse bias term

Figure 1 Out-of-sample prediction error and 95% confidence intervals of different estimators on synthetic data.

Results: Figure 1a shows results for the sparse bias term, while Figure 1b shows results for the non-sparse bias term (error bars represent 95% confidence intervals). We see that the joint estimator performs the best (excluding the oracle) in both the sparse and non-sparse cases. Sparsity does not appear to significantly affect the performance of any of the estimators except the joint estimator. In the sparse case, the joint estimator is very close to the oracle (i.e., we don't pay a significant price for not knowing the bias δ^*), since we are able to recover a very close approximation of δ^*. However, in the non-sparse case, the joint estimator yields only a slight improvement over the weighted estimator, and a sizeable gap remains between the oracle and the joint estimators (i.e., we pay a significant price for not knowing δ^*). Thus, the joint estimator successfully leverages sparsity when present, but still performs comparably or better than popular heuristics otherwise.

5.2. Recommendation Systems

Product variety has exploded, creating high search costs. As a consequence, many platforms offer data-driven recommendation systems that match customers with their preferred products. For example, Expedia is one of the world's largest online travel agencies, and serves millions of travelers a day. "In this competitive market matching users to hotel inventory is very important since users easily jump from website to website. As such, having the best ranking of hotels for specific users with the best integration of price competitiveness gives an [online travel agency] the best chance of winning the sale" (ICDM 2013). To inform these rankings, the goal is to train a model that can effectively predict which hotel rooms a customer will purchase.

In these settings, there are typically two outcomes: clicks and purchases. While purchases are the true outcome of interest, they are few and far between, making it hard to train an accurate model. On the other hand, clicks are much more frequent, and form a compelling proxy since customers will typically click on a product only if they have some intent to purchase. As a consequence, many recommendation systems use models that maximize click-through rates rather than purchase rates. In this case study, we take clicks and purchases as our proxy and true outcomes respectively.

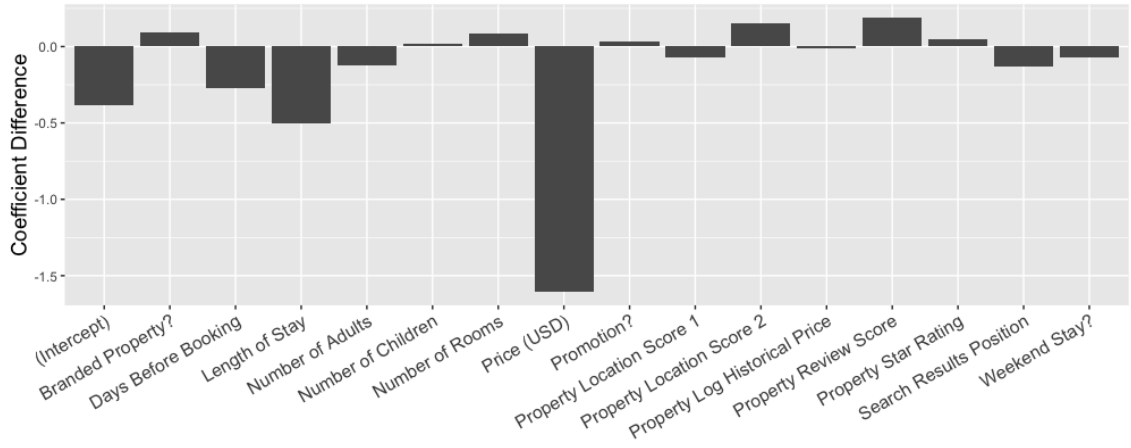


Figure 2 Difference in coefficients between a logistic regression predicting bookings and a logistic regression predicting clicks on the Expedia personalized recommendation dataset.

Data: We use personalized Expedia hotel search data that was made available through the 2013 International Conference on Data Mining challenge (ICDM 2013). We only consider the subset of data where search results were randomly sorted to avoid position bias of Expedia's recommendation algorithm. After pre-processing, there are over 2.2 million customer impressions, 15 customer- and hotel-specific features related to the search destination, and 2 outcomes (clicks and bookings). We note that 0.05% of impressions result in a click, while only 0.005% result in a purchase. Thus, the gold outcomes are an order of magnitude more sparse than the proxy outcomes.

More details on data pre-processing and model training are given in Appendix E.2.

Bias Term: Since we have a very large number of observations, we can train accurate logistic regression models[5] β^*_{proxy} and β^*_{gold} for clicks and bookings respectively. Fig 2 shows the difference in the resulting parameter estimates, i.e., $\delta^* = \beta^*_{gold} - \beta^*_{proxy}$. We immediately observe that the bias is in fact rather sparse — nearly all the coefficients of δ^* are negligible (absolute value of the coefficient is relatively close to 0), with the notable exception of the price coefficient. Thus, our assumption that δ^* is sparse appears well-founded on this data. Moreover, we observe a systematic bias, where the hotel price negatively impacts bookings far more than clicks. Intuitively, a customer may not mind browsing expensive travel products, but is unlikely to make an expensive purchase. Thus, using predicted click-through rates alone (i.e., the proxy estimator) to make recommendations could result in overly expensive recommendations, thereby hurting purchase rates.

Setup: In the data-rich regime (over 2 million impressions), the gold estimator is very accurate and there is no need for a proxy. We wish to study the data-scarce regime where proxies add value, so we restrict ourselves to small random subsamples of 10,000 impressions. Since we have binary

[5] We use logistic instead of linear regression since both outcomes are binary.

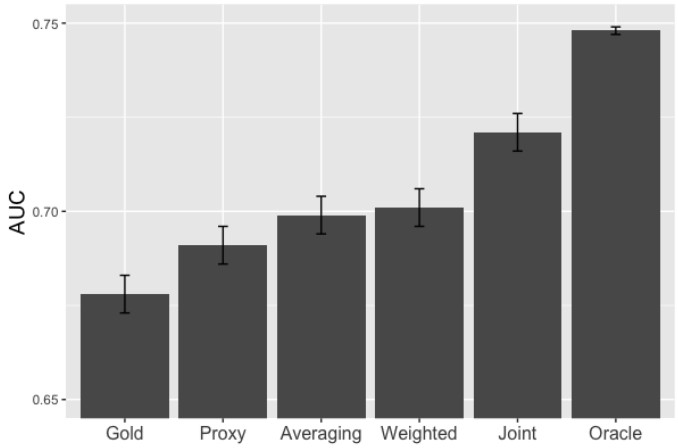


Figure 3 Predictive performance of different estimators on predicting bookings in the Expedia dataset.

outcomes, we use logistic rather than linear estimators. Note that $n_{gold} = n_{proxy}$, but the scarcity of bookings relative to clicks implies that $\sigma_{proxy} \ll \sigma_{gold}$. Similar to the previous subsection, the averaging, weighted, and joint estimators are trained on a training set, and their tuning parameters are optimized over a validation set. Our oracle is the gold estimator trained on the full Expedia dataset (over 2 million impressions) rather than the small subsample. Since we no longer have access to the true parameter, we use predictive performance on a held-out test set to assess the performance of our different estimators. Performance is measured by AUC (area under ROC curve), which is more reliable than accuracy[6] in imbalanced data (Friedman et al. 2001). We average our results over 100 trials, where we randomly draw our training and test sets in each iteration.

Results: Figure 3 shows the average performance on a held-out test set (error bars represent 95% confidence intervals). We see that the joint estimator performs the best (excluding the oracle). In particular, it bridges half the gap between the best baseline (weighted estimator) and the oracle. In roughly 70% of the trials, the joint estimator identifies price as a source of bias in $\hat{\delta}$.

5.3. Medical Risk Scoring

A key component of healthcare delivery is patient risk scoring. Identifying patients who are at risk for a particular adverse event can help inform early interventions and resource allocation. In this case study, we consider Type II diabetes. In 2012, approximately 8.3% of the world's adult population had diabetes, which is a leading cause of cardiovascular disease, renal disease, blindness, and limb amputation (Läll et al. 2017). To make matters worse, an estimated 40% of diabetics in the US are undiagnosed, placing them at risk for major health complications (Cowie et al. 2009). At the same time, several clinical trials have demonstrated the potential to prevent type II diabetes

[6] Accuracy is a poor metric for imbalanced data: a trivial estimator that always predicts 0 (no purchase) achieves 99.4% accuracy, since 99.4% of users do not purchase anything. A decision-maker would prefer an estimator that minimizes false positives while maintaining some baseline true positive rate (i.e., identifying most purchasing customers).

among high-risk individuals through lifestyle interventions (Tuomilehto et al. 2011). Thus, our goal is to accurately predict patient-specific risk for Type II diabetes to inform interventions.

There are typically two ways a healthcare provider can obtain a risk predictor: train a new risk predictor based on its own patient cohort (true cohort of interest), or use an existing risk predictor that has been trained on a different patient cohort (proxy cohort) at a different healthcare provider. Training a new model can bring with it data scarcity challenges for small- or medium-sized providers; on the other hand, implementing an existing model can be problematic due to differences in physician behavior, shifts in patient characteristics, and discrepancies from how data is encoded in the medical record. In this case study, we will take patient data from a medium-sized provider as our gold data, and patient data pooled from two larger providers as our proxy data.

Data: We use electronic medical record data across several healthcare providers. After basic pre-processing, we have roughly 100 features constructed from patient-specific information available *before* his/her most recent visit, and our outcome is an indicator variable for whether the patient was diagnosed with diabetes *during* his/her most recent visit. There are 980 patients in the proxy cohort (other providers), and 301 patients in the gold cohort (target provider), i.e., $n_{proxy} \gg n_{gold}$.

More details on data pre-processing and model training are given in Appendix E.3.

Setup: Once again we have binary outcomes, so we use logistic rather than linear estimators. Similar to the previous subsections, the averaging, weighted, and joint estimators are trained on a training set, and their tuning parameters are optimized over a validation set. Since we no longer have access to the true parameter, we use predictive performance on a held-out test set to assess the performance of our different estimators. Performance is measured by AUC (area under ROC curve), which is more reliable than accuracy[7] in imbalanced data (Friedman et al. 2001). We average our results over 100 trials, where we randomly draw our training and test sets in each iteration.

Results: Figure 4 shows the average performance on a held-out test set (error bars represent 95% confidence intervals). We see that the joint estimator performs the best by a significant margin.

Managerial Insights: The improved performance of the joint estimator suggests that there is systematic bias as play between the proxy and gold patient cohorts. A better understanding of these biases can provide valuable managerial insights, and help inform better feature engineering and/or improved models for risk prediction. Accordingly, we study the estimated bias term $\hat{\delta}$ across the 100 trials. We note that both the proxy and gold cohorts have similar rates of a diabetes diagnosis in the most recent visit: 14% and 13% respectively (the difference is not statistically significant).

[7] Again, due to data imbalance, a trivial estimator that always predicts 0 (low diabetes risk) achieves 87% accuracy, since 87% of patients will not be diagnosed with diabetes. A decision-maker would prefer an estimator that minimizes false positives while maintaining some baseline true positive rate (i.e., identifying most diabetic patients).

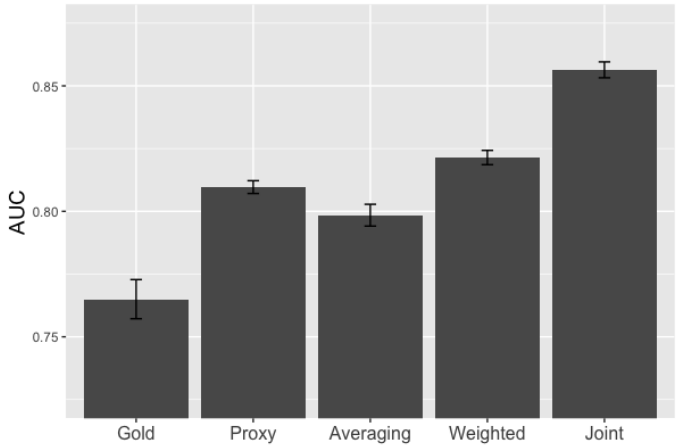


Figure 4 **Predictive performance of different estimators on predicting diabetes in a medical record dataset.**

One feature that is frequently identified in $\hat{\delta}$ is ICD-9 diagnosis code 790.21, which stands for "Impaired fasting glucose." Impaired fasting glucose (also known as pre-diabetes) occurs when blood glucose levels in the body are elevated during periods of fasting, and is an important indicator for diabetes risk. Despite having similar diabetes diagnosis rates across the proxy and gold cohorts, 4.6% of patients among the proxy cohort have an impaired fasting glucose diagnosis, while only 0.6% of patients among the gold cohort have this diagnosis. Conversations with a physician suggest that physicians at the target healthcare provider (gold cohort) may not wish to burden the patients with fasting, which is required to diagnose a patient with impaired fasting glucose; in contrast, physicians at the proxy healthcare providers appear more willing to do so. As a consequence, ICD-9 code 790.21 is a highly predictive feature in β^*_{proxy} for the proxy patient cohort, but not in β^*_{gold} for the gold patient cohort. Thus, differences in physician behavior can yield a systematic bias in the electronic medical records, and the joint estimator attempts to uncover such biases.

Similarly, another frequently identified feature is ICD-9 diagnosis code 278.0, which stands for "Overweight and obesity." Again, despite having similar diabetes diagnosis rates across the proxy and gold cohorts, 5.6% of patients among the proxy cohort have an obesity diagnosis, while only 0.9% of patients among the gold cohort have this diagnosis. However, there is no significant difference in the recorded patient BMIs across proxy and gold cohorts, suggesting that the difference in obesity *diagnosis* rates is not indicative of an actual difference in patient obesity rates. Conversations with a physician indicate that there are significant differences in how medical coders (staff responsible for encoding a patient's charts into the electronic medical record) choose which ICD-9 codes are recorded. As a consequence, ICD-9 code 278.0 is a highly predictive feature in β^*_{proxy} for the proxy patient cohort, but not in β^*_{gold} for the gold patient cohort. Thus, differences in how patient chart data is encoded in the medical record can also yield systematic biases, which the joint estimator attempts to uncover.

Apart from these examples, the estimated bias $\hat{\delta}$ also revealed differences in physician prescribing patterns. These biases are successfully leveraged by the joint estimator to improve performance.

6. Discussion and Conclusions

Proxies are copious and widely used in practice. However, the bias between the proxy predictive task and the true task of interest can lead to sub-optimal decisions. In this paper, we seek to *transfer* knowledge from proxies to the true task by imposing sparse structure on the bias between the two tasks. We propose a two-step estimator that uses techniques from high-dimensional statistics to efficiently combine a large amount of proxy data and a small amount of true data. Our estimator provably achieves the same accuracy as popular heuristics with up to exponentially less gold data.

Proxy data is often viewed as a means of improving predictive accuracy. However, even with infinite proxy samples, the proxy estimator's error is bounded below by its bias $\|\delta^*\|_1$. We propose that the true value of proxy data can actually lie in *enhancing* the value of gold data. For instance, consider the case where $n_{gold} \lesssim \mathcal{O}\left(d^2\sigma^2_{gold}\right)$. Our bounds show that the resulting gold OLS/ridge estimator's error is $\mathcal{O}(1)$. In other words, without proxy data, limited gold data offers no predictive value. Often, additional gold data can be very costly or impossible to obtain, explaining the frequent reliance on alternative (proxy) data sources. However, when we have sufficient proxy data, i.e., $n_{proxy} \gtrsim \mathcal{O}\left(d^2\sigma^2_{proxy}\right)$, we only require $\mathcal{O}\left(s^2 \log d \cdot \sigma^2_{gold}\right) \ll \mathcal{O}\left(d^2\sigma^2_{gold}\right)$ gold observations to improve estimation error. Thus, proxy data can help us more efficiently use gold data: instead of using the limited gold data directly for estimating the predictive model, our estimator uses gold data to efficiently de-bias the proxy estimator. This insight can inform experimental design, particularly when decision-makers trade off the costs for obtaining labeled proxy and gold data.

Recovering the bias term also yields important managerial insights. For instance, it can be very difficult for hospital management to discover the systematic differences in physician diagnosing behavior or data recording across hospitals. As discussed in Section 5, our estimator can recover an estimate of the bias term, which can shed light on the source of these biases. Once we understand these biases, one can perform better feature engineering, e.g., in the diabetes risk prediction example (Section 5.3), we may learn to use the BMI feature instead of the obesity diagnosis feature. Knowing the bias between the proxies may also help us identify better sources of proxy data, e.g., in medical risk prediction, we may try to use patient data from a hospital with diagnosing patterns that are more similar to those in the target hospital.

References

Ahsen, Mehmet Eren, Mehmet Ulvi Saygi Ayvaci, Srinivasan Raghunathan. 2018. When algorithmic predictions use human-generated data: A bias-aware classification algorithm for breast cancer diagnosis. *Information Systems Research* .

Axon, R Neal, Mark V Williams. 2011. Hospital readmission as an accountability measure. *Jama* **305**(5) 504–505.

Baardman, Lennart, Igor Levin, Georgia Perakis, Divya Singhvi. 2017. Leveraging comparables for new product sales forecasting .

Bastani, Hamsa, Osbert Bastani, Carolyn Kim. 2017. Interpreting predictive models for human-in-the-loop analytics .

Bayati, Mohsen, Sonia Bhaskar, Andrea Montanari. 2018. Statistical analysis of a low cost method for multiple disease prediction. *Statistical methods in medical research* **27**(8) 2312–2328.

Belloni, Alexandre, Daniel Chen, Victor Chernozhukov, Christian Hansen. 2012. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica* **80**(6) 2369–2429.

Belloni, Alexandre, Victor Chernozhukov, Christian Hansen. 2014. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies* **81**(2) 608–650.

Bickel, Peter, Ya'acov Ritov, Alexandre Tsybakov. 2009. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics* 1705–1732.

Brynjolfsson, Erik, Yu Jeffrey Hu, Mohammad S Rahman. 2013. *Competing in the age of omnichannel retailing*. MIT.

Bühlmann, Peter, Sara Van De Geer. 2011. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Candes, Emmanuel, Terence Tao. 2007. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics* 2313–2351.

Caruana, Rich. 1997. Multitask learning. *Machine learning* **28**(1) 41–75.

Chen, Kani, Inchi Hu, Zhiliang Ying, et al. 1999. Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs. *The Annals of Statistics* **27**(4) 1155–1163.

Chen, Scott S, David L Donoho, Michael A Saunders. 1995. Atomic decomposition by basis pursuit .

CMS. 2018. Readmissions reduction program (hrrp). Online. URL `https://www.cms.gov/medicare/medicare-fee-for-service-payment/acuteinpatientpps/readmissions-reduction-program.html`. [Last accessed October 2, 2018].

Collobert, Ronan, Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

Cowie, Catherine C, Keith F Rust, Earl S Ford, Mark S Eberhardt, Danita D Byrd-Holt, Chaoyang Li, Desmond E Williams, Edward W Gregg, Kathleen E Bainbridge, Sharon H Saydah, et al. 2009. Full accounting of diabetes and pre-diabetes in the us population in 1988–1994 and 2005–2006. *Diabetes care* **32**(2) 287–294.

Dzyabura, Daria, Srikanth Jagabathula, Eitan Muller. 2018. Accounting for discrepancies between online and offline product evaluations. *Marketing Science* .

Farias, Vivek F, Andrew A Li. 2017. Learning preferences with side information. *Management Science* .

Friedman, Jerome, Trevor Hastie, Robert Tibshirani. 2001. *The elements of statistical learning*, vol. 1. Springer series in statistics New York.

Homrighausen, Darren, Daniel J McDonald. 2014. Leave-one-out cross-validation is risk consistent for lasso. *Machine learning* **97**(1-2) 65–78.

ICDM. 2013. Personalized expedia hotel searches. URL `https://www.kaggle.com/c/expedia-personalized-sort`.

Jalali, Ali, Sujay Sanghavi, Chao Ruan, Pradeep K Ravikumar. 2010. A dirty model for multi-task learning. *Advances in neural information processing systems*. 964–972.

Läll, Kristi, Reedik Mägi, Andrew Morris, Andres Metspalu, Krista Fischer. 2017. Personalized risk prediction for type 2 diabetes: The potential of genetic risk scores. *Genetics in Medicine* **19**(3) 322.

Li, Ker-Chau, et al. 1987. Asymptotic optimality for c_p, c_l, cross-validation and generalized cross-validation: Discrete index set. *The Annals of Statistics* **15**(3) 958–975.

McCullagh, P., J. A. Nelder. 1989. *Generalized linear models (Second edition)*. London: Chapman & Hall.

Meier, Lukas, Sara Van De Geer, Peter Bühlmann. 2008. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(1) 53–71.

Milstein, Arnold. 2009. Ending extra payment for "never events"—stronger incentives for patients' safety. *New England Journal of Medicine* **360**(23) 2388–2390.

Mullainathan, Sendhil, Ziad Obermeyer. 2017. Does machine learning automate moral hazard and error? *American Economic Review* **107**(5) 476–80.

Negahban, Sahand, Bin Yu, Martin J Wainwright, Pradeep K Ravikumar. 2009. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *NIPS*. 1348–1356.

Nesterov, Yurii. 1998. Introductory lectures on convex programming volume i: Basic course .

Obermeyer, Ziad, Thomas H Lee. 2017. Lost in thought—the limits of the human mind and the future of medicine. *New England Journal of Medicine* **377**(13) 1209–1211.

Pan, Sinno Jialin, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **22**(10) 1345–1359.

Picard, Richard R, R Dennis Cook. 1984. Cross-validation of regression models. *Journal of the American Statistical Association* **79**(387) 575–583.

Raina, Rajat, Andrew Y Ng, Daphne Koller. 2006. Constructing informative priors using transfer learning. *Proceedings of the 23rd international conference on Machine learning*. ACM, 713–720.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Tsybakov, Alexander B, et al. 2004. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics* **32**(1) 135–166.

Tuomilehto, Jaakko, Peter Schwarz, Jaana Lindström. 2011. Long-term benefits from lifestyle interventions for type 2 diabetes prevention: time to expand the efforts. *Diabetes Care* **34**(Supplement 2) S210–S214.

Wainwright, Martin. 2016. *High-dimensional statistics: A non-asymptotic viewpoint*. Book Draft (Working Publication). URL `https://www.stat.berkeley.edu/~wainwrig/nachdiplom/Chap2_Sep10_2015.pdf`.

Zhang, Dennis, Hengchen Dai, Lingxiu Dong, Qian Wu, Lifan Guo, Xiaofei Liu. 2018. The value of pop-up stores in driving online engagement in platform retailing: Evidence from a large-scale field experiment with alibaba .

Appendix A: Baseline Estimators

We now prove lower bounds on the parameter estimation error for various heuristics (Theorems 1–4 from Section 3). Since we are considering worst-case error over allowable $\left\{\mathbf{X}_{gold}, \mathbf{X}_{proxy}, \beta^*_{gold}, \beta^*_{proxy}, \varepsilon_{gold}, \varepsilon_{proxy}\right\}$, it suffices to consider a simple example where the assumptions and bounds hold. Here, we consider

1. $\mathbf{X}_{gold}$ is chosen such that $\Sigma_{gold} = \frac{1}{n_{gold}}\mathbf{X}_{gold}^\top \mathbf{X}_{gold} = I_d$ (where I_d is the $d \times d$ identity matrix), and similarly, $\mathbf{X}_{proxy}$ is chosen such that $\Sigma_{proxy} = \frac{1}{n_{proxy}}\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy} = I_d$,
2. β^*_{gold} is chosen such that $\left\|\beta^*_{gold}\right\|_1 = 1$,
3. $\varepsilon_{gold} \sim \mathcal{N}(0, \sigma^2_{gold} I_d)$ and $\varepsilon_{proxy} \sim \mathcal{N}(0, \sigma^2_{proxy} I_d)$.

Clearly, all assumptions made in Section 2 hold for this case.

A.1. Proof of Theorem 1

Proof of Theorem 1 We first show a lower bound for the OLS estimator, followed by the ridge estimator.

(i) OLS Estimator: The OLS estimator has the well-known closed form expression $\hat{\beta}^{OLS}_{gold} = \left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold}\right)^{-1} \mathbf{X}_{gold}^\top Y_{gold}$. Plugging in for Y_{gold} and Σ_{gold} yields the (random) estimation error

$$\begin{aligned}\hat{\beta}^{OLS}_{gold} - \beta^*_{gold} &= \left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold}\right)^{-1} \mathbf{X}_{gold}^\top \varepsilon_{gold} \\ &= \frac{1}{n_{gold}} \mathbf{X}_{gold}^\top \varepsilon_{gold}\,.\end{aligned}$$

Then, we can compute the variance

$$\begin{aligned}\mathrm{Var}\left(\hat{\beta}^{OLS}_{gold} - \beta^*_{gold}\right) &= \frac{1}{n^2_{gold}} \mathbb{E}\left[\mathbf{X}_{gold}^\top \varepsilon_{gold} \varepsilon_{gold}^\top \mathbf{X}_{gold}\right] \\ &= \frac{\sigma^2_{gold}}{n_{gold}}\,.\end{aligned}$$

Thus, using the distribution of ε_{gold}, we can write

$$\hat{\beta}^{OLS}_{gold} - \beta^*_{gold} \sim \mathcal{N}\left(0, \frac{\sigma^2_{gold}}{n_{gold}} I_d\right)\,.$$

Applying Lemma 10 in Appendix D, it follows that

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\beta}^{OLS}_{gold} - \beta^*_{gold}\right\|_1\right] &= \mathrm{tr}\left(I_d\right)\sqrt{\frac{2\sigma^2}{\pi n_{gold}}} \\ &= d\sqrt{\frac{2\sigma^2}{\pi n_{gold}}}\,.\end{aligned}$$

This computation gives us a lower bound of the parameter estimation error for the OLS estimator.

(ii) Ridge Estimator: Next, we consider the ridge estimator, which has the well-known closed form expression $\hat{\beta}^{ridge}_{gold}(\lambda) = \left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold} + \lambda I_d\right)^{-1} \mathbf{X}_{gold}^\top Y_{gold}$. Plugging in for Y_{gold} and Σ_{gold} yields the (random) estimation error

$$\begin{aligned}\hat{\beta}^{ridge}_{gold}(\lambda) - \beta^*_{gold} &= \left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold} + \lambda I_d\right)^{-1} \mathbf{X}_{gold}^\top \mathbf{X}_{gold} \beta^*_{gold} + \left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold} + \lambda I_d\right)^{-1} \mathbf{X}_{gold}^\top \varepsilon_{gold} - \beta^*_{gold} \\ &= \frac{n_{gold}}{n_{gold} + \lambda}\beta^*_{gold} + \frac{1}{n_{gold} + \lambda}\mathbf{X}_{gold}^\top \varepsilon_{gold} - \beta^*_{gold} \\ &= -\frac{\lambda}{n_{gold} + \lambda}\beta^*_{gold} + \frac{1}{n_{gold} + \lambda}\mathbf{X}_{gold}^\top \varepsilon_{gold}\,.\end{aligned}$$

Then, we can compute the variance (note that the true parameter β^*_{gold} is not a random variable)

$$\begin{aligned}\mathrm{Var}\left(\hat{\beta}^{ridge}_{gold}(\lambda)-\beta^*_{gold}\right) &= \frac{1}{(n_{gold}+\lambda)^2}\mathbb{E}\left[\mathbf{X}^\top_{gold}\varepsilon_{gold}\varepsilon^\top_{gold}\mathbf{X}_{gold}\right]\\ &= \frac{n_{gold}\sigma^2_{gold}}{(n_{gold}+\lambda)^2}\,.\end{aligned}$$

Thus, using the distribution of ε_{gold}, we can write

$$\hat{\beta}^{ridge}_{gold}(\lambda)-\beta^*_{gold}\sim\mathcal{N}\left(-\frac{\lambda}{n_{gold}+\lambda}\beta^*_{gold},\frac{n_{gold}\sigma^2_{gold}}{(n_{gold}+\lambda)^2}I_d\right).$$

Applying Lemma 11 in Appendix D, it follows that

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\beta}^{ridge}_{gold}(\lambda)-\beta^*_{gold}\right\|_1\right] &\geq \max\left\{\frac{1}{2}\cdot\frac{\lambda}{n_{gold}+\lambda}\left\|\beta^*_{gold}\right\|_1,\sqrt{\frac{n_{gold}\sigma^2_{gold}}{2\pi(n_{gold}+\lambda)^2}}\,\mathrm{tr}\left(I_d\right)\right\}\\ &= \max\left\{\frac{1}{2}\cdot\frac{\lambda b}{n_{gold}+\lambda},d\sqrt{\frac{n_{gold}\sigma^2_{gold}}{2\pi(n_{gold}+\lambda)^2}}\right\}.\end{aligned}$$

Note that the first term in the maximum is monotone increasing in λ, while the second term in the maximum is monotone decreasing in λ. Thus, the minimum value of the maximum is achieved when the two terms are equal, i.e., when

$$\lambda=\frac{d}{b}\sqrt{\frac{2n_{gold}\sigma^2_{gold}}{\pi}}\,.$$

Plugging in, we get

$$\min_{\lambda}\ \mathbb{E}\left[\left\|\hat{\beta}^{ridge}_{gold}(\lambda)-\beta^*_{gold}\right\|_1\right]\geq\frac{d\sigma_{gold}/\sqrt{2\pi}}{b\sqrt{n_{gold}}+d\sigma_{gold}\sqrt{2/\pi}}\,.$$

Along with the previous case, this completes the proof. □

A.2. Proof of Theorem 2

Proof of Theorem 2 Note that the OLS estimator is $\hat{\beta}_{proxy}=\left(\mathbf{X}^\top_{proxy}\mathbf{X}_{proxy}\right)^{-1}\mathbf{X}^\top_{proxy}Y_{proxy}$. Plugging in for Y_{proxy} and Σ_{proxy} yields the (random) estimation error

$$\begin{aligned}\hat{\beta}_{proxy}-\beta^*_{proxy} &= \left(\mathbf{X}^\top_{proxy}\mathbf{X}_{proxy}\right)^{-1}\mathbf{X}^\top_{proxy}\varepsilon_{proxy}\\ &= \frac{1}{n_{proxy}}\mathbf{X}^\top_{proxy}\varepsilon_{proxy}\,.\end{aligned}$$

Then, we can compute the variance

$$\begin{aligned}\mathrm{Var}\left(\hat{\beta}_{proxy}-\beta^*_{proxy}\right) &= \frac{1}{n^2_{proxy}}\mathbb{E}\left[\mathbf{X}^\top_{proxy}\varepsilon_{proxy}\varepsilon^\top_{proxy}\mathbf{X}_{proxy}\right]\\ &= \frac{\sigma^2_{proxy}}{n_{proxy}}\,.\end{aligned}$$

Thus, using the distribution of ε_{gold} and the fact that $\beta^*_{gold}=\beta^*_{proxy}+\delta^*$, we can write

$$\hat{\beta}_{proxy}-\beta^*_{gold}\sim\mathcal{N}\left(-\delta^*,\frac{\sigma^2_{proxy}}{n_{proxy}}I_d\right).$$

Applying Lemma 11 in Appendix D, it follows that

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\beta}^{OLS}_{gold}-\beta^*_{gold}\right\|_1\right] &\geq \max\left\{\frac{1}{2}\left\|\delta^*\right\|_1,\sqrt{\frac{\sigma^2_{proxy}}{2\pi n_{proxy}}}\,\mathrm{tr}\left(I_d\right)\right\}\\ &= \max\left\{\frac{1}{2}\left\|\delta^*\right\|_1,d\sqrt{\frac{\sigma^2_{proxy}}{2\pi n_{proxy}}}\right\}.\end{aligned}$$

This computation gives us a lower bound of the parameter estimation error for the OLS estimator. □

A.3. Proof of Theorem 3

Proof of Theorem 3 The averaging estimator can be expanded as

$$\begin{aligned}
\hat{\beta}_{avg}(\lambda) &= (1-\lambda)\cdot\hat{\beta}_{gold}^{OLS} + \lambda\cdot\hat{\beta}_{proxy} \\
&= (1-\lambda)\cdot\left(\mathbf{X}_{gold}^{\top}\mathbf{X}_{gold}\right)^{-1}\mathbf{X}_{gold}^{\top}Y_{gold} + \lambda\cdot\left(\mathbf{X}_{proxy}^{\top}\mathbf{X}_{proxy}\right)^{-1}\mathbf{X}_{proxy}^{\top}Y_{proxy} \\
&= \frac{1-\lambda}{n_{gold}}\cdot\mathbf{X}_{gold}^{\top}Y_{gold} + \frac{\lambda}{n_{proxy}}\cdot\mathbf{X}_{proxy}^{\top}Y_{proxy} \\
&= \frac{1-\lambda}{n_{gold}}\cdot\mathbf{X}_{gold}^{\top}\left(\mathbf{X}_{gold}\beta_{gold}^{*} + \varepsilon_{gold}\right) + \frac{\lambda}{n_{proxy}}\cdot\mathbf{X}_{proxy}^{\top}\left(\mathbf{X}_{proxy}\beta_{proxy}^{*} + \varepsilon_{proxy}\right) \\
&= (1-\lambda)\beta_{gold}^{*} + \lambda\beta_{proxy}^{*} + \frac{1-\lambda}{n_{gold}}\cdot\mathbf{X}_{gold}^{\top}\varepsilon_{gold} + \frac{\lambda}{n_{proxy}}\cdot\mathbf{X}_{proxy}^{\top}\varepsilon_{proxy}\,.
\end{aligned}$$

Then,

$$\hat{\beta}_{avg}(\lambda) - \beta_{gold}^{*} = \lambda\left(\beta_{proxy}^{*} - \beta_{gold}^{*}\right) + \frac{1-\lambda}{n_{gold}}\cdot\mathbf{X}_{gold}^{\top}\varepsilon_{gold} + \frac{\lambda}{n_{proxy}}\cdot\mathbf{X}_{proxy}^{\top}\varepsilon_{proxy}\,.$$

Using the fact that ε_{gold} and ε_{proxy} are independent random variables, we can compute

$$\begin{aligned}
\operatorname{Var}\left(\hat{\beta}_{avg}(\lambda) - \beta_{gold}^{*}\right) &= \left(\frac{1-\lambda}{n_{gold}}\right)^{2}\operatorname{Var}\left(\mathbf{X}_{gold}^{\top}\varepsilon_{gold}\right) + \left(\frac{\lambda}{n_{proxy}}\right)^{2}\operatorname{Var}\left(\mathbf{X}_{proxy}^{\top}\varepsilon_{proxy}\right) \\
&= \left(\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{n_{gold}} + \frac{\lambda^{2}\sigma_{proxy}^{2}}{n_{proxy}}\right)I_{d}\,.
\end{aligned}$$

Thus, we have that

$$\hat{\beta}_{avg}(\lambda) - \beta_{gold}^{*} \sim \mathcal{N}\left(\lambda\left(\beta_{proxy}^{*} - \beta_{gold}^{*}\right), \left(\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{n_{gold}} + \frac{\lambda^{2}\sigma_{proxy}^{2}}{n_{proxy}}\right)I_{d}\right).$$

Applying Lemma 11 in Appendix D, it follows that

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\beta}_{avg} - \beta_{gold}^{*}\right\|_{1}\right] &\geq \max\left\{\frac{\lambda}{2}\left\|\beta_{proxy}^{*} - \beta_{gold}^{*}\right\|_{1}, \sqrt{\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{2\pi n_{gold}} + \frac{\lambda^{2}\sigma_{proxy}^{2}}{2\pi n_{proxy}}}\,\operatorname{tr}\left(I_{d}\right)\right\} \\
&= \max\left\{\frac{\lambda}{2}\left\|\delta^{*}\right\|_{1}, d\sqrt{\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{2\pi n_{gold}} + \frac{\lambda^{2}\sigma_{proxy}^{2}}{2\pi n_{proxy}}}\right\} \\
&\geq \max\left\{\frac{\lambda}{2}\left\|\delta^{*}\right\|_{1}, d\sqrt{\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{2\pi n_{gold}}}, d\sqrt{\frac{\lambda^{2}\sigma_{proxy}^{2}}{2\pi n_{proxy}}}\right\} \\
&\geq \frac{\lambda}{6}\left\|\delta^{*}\right\|_{1} + \frac{d}{3}\sqrt{\frac{\lambda^{2}\sigma_{proxy}^{2}}{2\pi n_{proxy}}} + \frac{d}{3}\sqrt{\frac{(1-\lambda)^{2}\sigma_{gold}^{2}}{2\pi n_{gold}}}\,,
\end{aligned}$$

where we have used the identity $\max\{a,b\} \geq \frac{a+b}{2}$. Now, note that this expression is linear in λ, and thus, the value of λ that minimizes the maximum occurs at one of the two extrema, i.e., $\lambda = 0$ or $\lambda = 1$. Then, we can write

$$\min_{\lambda}\ \mathbb{E}\left[\left\|\hat{\beta}_{avg} - \beta_{gold}^{*}\right\|_{1}\right] \geq \min\left\{\frac{d\sigma_{gold}}{3\sqrt{2\pi n_{gold}}}, \frac{1}{6}\left\|\delta^{*}\right\|_{1} + \frac{d\sigma_{proxy}}{3\sqrt{2\pi n_{proxy}}}\right\}.$$

□

A.4. Proof of Theorem 4

Proof of Theorem 4 Recall that the weighted estimator is given by

$$\hat{\beta}_{weight}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{\lambda n_{gold} + n_{proxy}} \cdot \left(\lambda \|Y_{gold} - \mathbf{X}_{gold}\beta\|_2^2 + \|Y_{proxy} - \mathbf{X}_{proxy}\beta\|_2^2 \right) \right\}, .$$

Setting the gradient to 0, we get that $\hat{\beta}_{weight}(\lambda)$ is the solution to the equation

$$\frac{2\lambda}{\lambda n_{gold} + n_{proxy}} \cdot \mathbf{X}_{gold}^\top (Y_{gold} - \mathbf{X}_{gold}\beta) + \frac{2}{\lambda n_{gold} + n_{proxy}} \cdot \mathbf{X}_{proxy}^\top (Y_{proxy} - \mathbf{X}_{proxy}\beta) = 0 \,.$$

It is useful to define the variable

$$\lambda' = \frac{n_{proxy}}{\lambda n_{gold} + n_{proxy}} \,,$$

and observe that

$$1 - \lambda' = \frac{\lambda n_{gold}}{\lambda n_{gold} + n_{proxy}} \,.$$

Note that the allowed range of $\lambda \in [0, \infty)$ corresponds to the allowed range $\lambda' \in [0, 1]$. Thus, we can equivalently write $\hat{\beta}_{weight}$ as a function of $\lambda' \in [0, 1]$ is the solution to

$$\frac{1-\lambda'}{n_{gold}} \cdot \mathbf{X}_{gold}^\top (Y_{gold} - \mathbf{X}_{gold}\beta) + \frac{\lambda'}{n_{proxy}} \cdot \mathbf{X}_{proxy}^\top (Y_{proxy} - \mathbf{X}_{proxy}\beta) = 0 \,,$$

which yields the solution

$$\begin{aligned}
\hat{\beta}_{weight}(\lambda') &= \left(\frac{1-\lambda'}{n_{gold}} \cdot \mathbf{X}_{gold}^\top \mathbf{X}_{gold} + \frac{\lambda'}{n_{proxy}} \cdot \mathbf{X}_{proxy}^\top \mathbf{X}_{proxy} \right)^{-1} \left(\frac{1-\lambda'}{n_{gold}} \cdot \mathbf{X}_{gold}^\top Y_{gold} + \frac{\lambda'}{n_{proxy}} \cdot \mathbf{X}_{proxy}^\top Y_{proxy} \right) \\
&= \frac{1-\lambda'}{n_{gold}} \cdot \mathbf{X}_{gold}^\top Y_{gold} + \frac{\lambda'}{n_{proxy}} \cdot \mathbf{X}_{proxy}^\top Y_{proxy} \,.
\end{aligned}$$

Note that this expression is exactly the averaging estimator (Section 3.3) in this setting, and thus the proof and lower bound of Theorem 3 apply directly. This completes the proof. □

Appendix B: Linear Joint Estimator

B.1. Missing Lemmas in Proof of Theorem 5

Proof of Lemma 2 Recall that

$$\nu \;=\; \hat{\beta}_{proxy} - \beta^*_{proxy} \;=\; \left(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy} \right)^{-1} \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \,.$$

Then, on event $\mathcal{I}$, we can write

$$\begin{aligned}
\|\mathbf{X}_{gold}\nu\|_2^2 &= \left\| \mathbf{X}_{gold} (\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1} \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \\
&\le \|\mathbf{X}_{gold}\|_2^2 \cdot \left\| (\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1} \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \\
&\le \frac{1}{\psi^2 n_{proxy}^2} \|\mathbf{X}_{gold}\|_2^2 \cdot \left\| \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \\
&\le \frac{d n_{gold}}{\psi^2 n_{proxy}^2} \left\| \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \\
&\le \frac{d n_{gold}}{\psi^2 n_{proxy}^2} \lambda_1 \,,
\end{aligned}$$

where the first inequality follows from the definition of the matrix norm, the second inequality follows from Assumption 2 on the minimum eigenvalue of Σ_{proxy} yielding $\left(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy}\right)^{-1} = \frac{1}{n_{proxy}} \Sigma_{proxy}^{-1} \preceq \frac{1}{\psi n_{proxy}} I_d$, and the third inequality follows from the matrix norm identity that

$$\|\mathbf{X}_{gold}\|_2^2 \;\leq\; \operatorname{tr}\left(\mathbf{X}_{gold}^\top \mathbf{X}_{gold}\right) \;=\; n_{gold} \operatorname{tr}\left(\Sigma_{gold}\right) \;=\; d n_{gold}\,,$$

using the fact that we normalized $\operatorname{diag}\left(\Sigma_{gold}\right) = 1_{d\times 1}$. This proves the first inequality.

For the second inequality, we observe that on event $\mathcal{I}$,

$$\begin{aligned}
\|\nu\|_1 &= \left\|(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1} \mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_1 \\
&\leq \sqrt{d} \left\|(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1} \mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2 \\
&\leq \sqrt{d} \left\|(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1}\right\|_2 \cdot \left\|\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2 \\
&\leq \frac{\sqrt{d}}{\psi n_{proxy}} \left\|\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2 \\
&\leq \frac{\sqrt{d\lambda_1}}{\psi n_{proxy}}\,,
\end{aligned}$$

where the first inequality follows from the definition of the matrix norm, the second inequality follows from Cauchy Schwarz, and the third inequality follows from Assumption 2. □

Proof of Lemma 4 Recall from Eq. (5) that $\mathcal{J} = \left\{ \frac{2}{n_{gold}} \left\|\varepsilon_{gold}^\top \mathbf{X}_{gold}\right\|_\infty \leq \lambda_0 \right\}$. Then,

$$\begin{aligned}
\Pr\left[\mathcal{J}\right] &= 1 - \Pr\left[\max_{r\in[d]} \left|\varepsilon_{gold}^\top \mathbf{X}_{gold}^{(r)}\right| \;\geq\; \frac{\lambda_0 n_{gold}}{2}\right] \\
&\geq 1 - d \cdot \Pr\left[\left|\varepsilon_{gold}^\top \mathbf{X}_{gold}^{(1)}\right| \;\geq\; \frac{\lambda_0 n_{gold}}{2}\right],
\end{aligned}$$

where $\mathbf{X}_{gold}^{(r)} \in \mathbb{R}^d$ is the r^{th} column of $\mathbf{X}_{gold}$, and the first inequality follows from a union bound.

We can then apply Lemma 15 in Appendix D with $z_i = (\varepsilon_{gold})_i$, $a_i = (\mathbf{X}_{gold})_{i,r}$, noting that our standardization of $\mathbf{X}_{gold}$ implies each column satisfies

$$A = \sum_{i=1}^{n_{gold}} (\mathbf{X}_{gold})_{i,r}^2 = \left\|\mathbf{X}_{gold}^{(r)}\right\|_2^2 = n_{gold}\,.$$

Then, by Lemma 15,

$$W = \sum_{i=1}^{n_{gold}} (\mathbf{X}_{gold})_{i,r} (\varepsilon_{gold})_i = \varepsilon_{gold}^\top \mathbf{X}_{gold}^{(r)}\,,$$

is a $\left(\sigma_{gold}\sqrt{n_{gold}}\right)$-subgaussian random variable. Thus, we can apply a concentration inequality for subgaussian random variables (Lemma 13 in Appendix D) to bound

$$\begin{aligned}
\Pr\left[\mathcal{J}\right] &\geq 1 - d \cdot \Pr\left[\left|\varepsilon_{gold}^\top \mathbf{X}_{gold}^{(1)}\right| \;\geq\; \frac{\lambda_0 n_{gold}}{2}\right] \\
&\geq 1 - 2d \exp\left(-\frac{\lambda_0^2 n_{gold}}{8\sigma_{gold}^2}\right).
\end{aligned}$$

□

Proof of Lemma 5 Recall from Eq. (6) that $\mathcal{I} = \left\{ \left\| \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \leq \lambda_1 \right\}$. Then,

$$\begin{aligned} \left\| \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 &= \left\| \sum_{i=1}^{n_{proxy}} \left(\mathbf{X}_{proxy}\right)_i \left(\varepsilon_{proxy}\right)_i \right\|_2^2 \\ &= \sum_{j=1}^{d} \left(\sum_{i=1}^{n_{proxy}} \left(\mathbf{X}_{proxy}\right)_{i,j} \varepsilon_{proxy}^{(i)} \right)^2 , \end{aligned}$$

where $\left(\mathbf{X}_{proxy}\right)_i \in \mathbb{R}^d$ is the i^{th} row of $\mathbf{X}_{proxy}$, and $\left(\varepsilon_{proxy}\right)_i \in \mathbb{R}$ is the i^{th} component of ε_{proxy}. Note that $\left\{\varepsilon_{proxy}^{(i)}\right\}_{i=1}^{n_{proxy}}$ are independent σ_{proxy}-subgaussian random variables. Thus, each summand is a linear combination of independent subgaussian random variables, which yields a new subgaussian random variable. For each $j \in [d]$, it is useful to define an intermediate variable

$$\varepsilon'_j = \sum_{i=1}^{n_{proxy}} \left(\mathbf{X}_{proxy}^{(i)}\right)_j \varepsilon_{proxy}^{(i)}.$$

Note that $\mathbf{X}_{proxy}^\top \varepsilon_{proxy} \in \mathbb{R}^{d \times 1}$ is a vector whose elements are ε'_j. We can now apply Lemma 15 in Appendix D, taking $\{z_i\}$ to be $\left\{\varepsilon_{proxy}^{(i)}\right\}$, $\{a_i\}$ to be $\left\{\left(\mathbf{X}_{proxy}^{(i)}\right)_j\right\}$, and noting that

$$A = \sum_{i=1}^{n_{proxy}} \left(\mathbf{X}_{proxy}^{(i)}\right)_j \left(\mathbf{X}_{proxy}^{(i)}\right)_j \;=\; n_{proxy} \Sigma_{proxy}^{(jj)} \;=\; n_{proxy}\,,$$

for each $j \in [d]$ since we have normalized $\operatorname{diag}\left(\Sigma_{proxy}\right) = 1_{d \times 1}$. Then, by Lemma 15, ε'_j is $\left(\sigma_{proxy}\sqrt{n_{proxy}}\right)$-subgaussian. We can then apply a concentration inequality for subgaussian random variables (Lemma 13 in Appendix D) to bound

$$\begin{aligned} \Pr[\mathcal{I}] &= 1 - \Pr\left[\left\| \mathbf{X}_{proxy}^\top \varepsilon_{proxy} \right\|_2^2 \;\geq\; \lambda_1 \right] \\ &= 1 - \Pr\left[\left\| \varepsilon' \right\|_2^2 \;\geq\; \lambda_1 \right] \\ &\geq 1 - d \cdot \Pr\left[\left| \varepsilon'_j \right| \;\geq\; \sqrt{\frac{\lambda_1}{d}} \right] \\ &\geq 1 - 2d \exp\left(-\frac{\lambda_1}{2d\sigma_{proxy}^2 n_{proxy}} \right) . \end{aligned}$$

□

B.2. Proof of Corollary 1

Proof of Corollary 1 For ease of notation, let

$$w = \frac{5}{4\psi^2} + \frac{5}{\psi} + \frac{5s}{4\phi^2}\,.$$

Recall from Lemma 3 that

$$\left\| \hat{\beta}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\leq\; \lambda w\,,$$

with probability 1 when the events $\mathcal{J}$ and $\mathcal{I}$ hold, and we take $\lambda_0 = \lambda/5$ and $\lambda_1 = n_{proxy}^2 \lambda^2 / d$.

We expand the expected parameter estimation error

$$\begin{aligned} \mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \right] &= \mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[\mathcal{J} \cap \mathcal{I}] \\ &\quad + \mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; \mathcal{J}^C \cup \mathcal{I}^C \right] \cdot \Pr[\mathcal{J}^C \cup \mathcal{I}^C]\,. \end{aligned} \tag{20}$$

To bound the first expectation on the right hand side of (20), we define a new event

$$B = \left(\left\| \hat{\beta}_{joint}(\lambda) \right\|_1 \leq 2b \right).$$

By definition, $\hat{\beta}^{tr}_{joint} = \hat{\beta}_{joint}$ when B holds, and $\hat{\beta}^{tr}_{joint} = 0$ otherwise. Then,

$$\begin{aligned}
&\mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; \mathcal{J} \cap \mathcal{I} \right] \\
&= \mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; B \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B] + \mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; B^C \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B^C] \\
&= \mathbb{E}\left[\left\| \hat{\beta}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; B \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B] + \mathbb{E}\left[\left\| \beta^*_{gold} \right\|_1 \;\middle|\; B^C \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B^C] \\
&\leq \lambda w \cdot \Pr[B] + \mathbb{E}\left[\left\| \beta^*_{gold} \right\|_1 \mid B^C \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B^C].
\end{aligned}$$

Now, note that on the event $(B^C \cap \mathcal{J} \cap \mathcal{I})$, we have both that

$$\begin{aligned}
\left\| \hat{\beta}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 &\leq \lambda w, \\
\left\| \hat{\beta}_{joint}(\lambda) \right\|_1 &\geq 2b \geq 2 \left\| \beta^*_{gold} \right\|_1.
\end{aligned}$$

Together, these facts imply that on the event $(B^C \cap \mathcal{J} \cap \mathcal{I})$,

$$\begin{aligned}
\left\| \beta^*_{gold} \right\|_1 &\leq \left\| \hat{\beta}_{joint}(\lambda) \right\|_1 - \left\| \beta^*_{gold} \right\|_1 \\
&\leq \left\| \hat{\beta}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \\
&\leq \lambda w,
\end{aligned}$$

using the triangle inequality. Thus,

$$\begin{aligned}
\mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \;\middle|\; \mathcal{J} \cap \mathcal{I} \right] &\leq \lambda w \cdot \Pr[B] + \mathbb{E}\left[\left\| \beta^*_{gold} \right\|_1 \mid B^C \cap \mathcal{J} \cap \mathcal{I} \right] \cdot \Pr[B^C] \\
&\leq \lambda w \cdot \Pr[B] + \lambda w \cdot \Pr[B^C] \\
&= \lambda w. \qquad (21)
\end{aligned}$$

Next, we consider the second expectation on the right hand side of (20). Regardless of the events $\mathcal{J}, \mathcal{I}$, and B, we always have the following bound

$$\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \leq \left\| \hat{\beta}^{tr}_{joint}(\lambda) \right\|_1 + \left\| \beta^*_{gold} \right\|_1 \leq 3b, \qquad (22)$$

using the triangle inequality and the definition of $\hat{\beta}^{tr}_{joint}(\lambda)$. Substituting (21) and (22) into (20), we have

$$\begin{aligned}
\mathbb{E}\left[\left\| \hat{\beta}^{tr}_{joint}(\lambda) - \beta^*_{gold} \right\|_1 \right] &\leq \lambda w \cdot \Pr[\mathcal{J} \cap \mathcal{I}] + 3b \cdot \Pr[\mathcal{J}^C \cup \mathcal{I}^C] \\
&\leq \lambda w + 3b \cdot \left(\Pr[J^C] + \Pr[\mathcal{I}^C] \right) \\
&\leq \lambda w + 6bd \cdot \left(\exp\left(-\frac{\lambda^2 n_{gold}}{200 \sigma^2_{gold}} \right) + \exp\left(-\frac{\lambda^2 n_{proxy}}{2 d^2 \sigma^2_{proxy}} \right) \right),
\end{aligned}$$

using a union bound, and applying Lemmas 4 and 5 with the chosen values $\lambda_0 = \lambda/5$ and $\lambda_1 = n^2_{proxy} \lambda^2 / d$.

□

Appendix C: Nonlinear Joint Estimator

Throughout this section, we use the same notation as we did in the linear case. By definition,

$$\beta^*_{gold} = \hat{\beta}_{proxy} + \min_{\delta} \mathbb{E}_{\varepsilon_{gold}} \mathcal{L}(\delta) = \hat{\beta}_{proxy} + \tilde{\delta}.$$

Thus, defining $\nu = \hat{\beta}_{proxy} - \beta^*_{proxy}$ as before, we have $\tilde{\delta} = \delta^* - \nu$.

C.1. Key Lemmas for GLMs

Proof of Lemma 6 Using the definition of the empirical log likelihood $\mathcal{L}$, we can expand

$$\begin{aligned}\mathcal{E}(\delta) &= \mathbb{E}_{\varepsilon_{gold}}\left[\mathcal{L}(\delta) - \mathcal{L}(\tilde{\delta})\right] \\ &= \frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \left\{ -A'\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\left(\delta - \tilde{\delta}\right)^\top \mathbf{X}^{(i)}_{gold} + A\left(\left(\delta + \hat{\beta}_{proxy}\right)^\top \mathbf{X}^{(i)}_{gold}\right) - A\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\right\}.\end{aligned}$$

We can apply Assumption 4 and Lemma 16 in Appendix D to bound the second and third term as

$$\begin{aligned}A\left(\left(\delta + \hat{\beta}_{proxy}\right)^\top \mathbf{X}^{(i)}_{gold}\right) - A\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right) \geq A'\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\left(\delta - \tilde{\delta}\right)^\top \mathbf{X}^{(i)}_{gold} \\ + \frac{m}{2}\left\|\mathbf{X}^{(i)}_{gold}\left(\delta - \tilde{\delta}\right)\right\|_2^2.\end{aligned}$$

Substituting the above yields

$$\begin{aligned}\mathcal{E}(\delta) &\geq \frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \frac{m}{2}\left\|\mathbf{X}^{(i)}_{gold}\left(\delta - \tilde{\delta}\right)\right\|_2^2 \\ &\geq \frac{m}{2n_{gold}}\left\|\mathbf{X}_{gold}\left(\delta - \tilde{\delta}\right)\right\|_2^2.\end{aligned}$$

□

LEMMA 7. *The empirical process can be bounded as*

$$\left|w(\hat{\delta}) - w(\tilde{\delta})\right| \;\leq\; \frac{1}{n_{gold}}\left\|\hat{\delta} - \tilde{\delta}\right\|_1 \cdot \left\|\varepsilon^\top_{gold}\mathbf{X}_{gold}\right\|_\infty.$$

Proof of Lemma 7 First, expanding the loss function and substituting $\tilde{\delta} + \hat{\beta}_{proxy} = \beta^*_{gold}$ yields

$$\mathcal{L}(\hat{\delta}) - \mathcal{L}(\tilde{\delta}) = \frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \left\{ -Y^{(i)}_{gold}\left(\hat{\delta} - \tilde{\delta}\right)^\top \mathbf{X}^{(i)}_{gold} + A\left(\left(\hat{\delta} + \hat{\beta}_{proxy}\right)^\top \mathbf{X}^{(i)}_{gold}\right) - A\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\right\}.$$

Second, noting that $\mathbb{E}_{\varepsilon_{gold}} Y^{(i)}_{gold} = A'\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)$, we can write

$$\begin{aligned}\mathbb{E}_{\varepsilon_{gold}}\left[\mathcal{L}(\hat{\delta}) - \mathcal{L}(\tilde{\delta})\right] = \frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \Bigg\{ -A'\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\left(\hat{\delta} - \tilde{\delta}\right)^\top \mathbf{X}^{(i)}_{gold} \\ + A\left(\left(\hat{\delta} + \hat{\beta}_{proxy}\right)^\top \mathbf{X}^{(i)}_{gold}\right) + A\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\Bigg\}.\end{aligned}$$

Combining these expressions, we get

$$\begin{aligned}\left|w(\hat{\delta}) - w(\tilde{\delta})\right| &= \left|\frac{1}{n_{gold}} \sum_{i=1}^{n_{gold}} \left\{\left(-Y^{(i)}_{gold} + A'\left(\left(\beta^*_{gold}\right)^\top \mathbf{X}^{(i)}_{gold}\right)\right)\left(\hat{\delta} - \tilde{\delta}\right)^\top \mathbf{X}^{(i)}_{gold}\right\}\right| \\ &= \left|\left(\hat{\delta} - \tilde{\delta}\right)^\top \left(\sum_{i=1}^{n_{gold}} \mathbf{X}^{(i)}_{gold}\varepsilon^{(i)}_{gold}\right)\right| \\ &= \left|\left(\hat{\delta} - \tilde{\delta}\right)^\top \mathbf{X}^\top_{gold}\varepsilon_{gold}\right| \\ &\leq \frac{1}{n_{gold}}\left\|\hat{\delta} - \tilde{\delta}\right\|_1 \cdot \left\|\varepsilon^\top_{gold}\mathbf{X}_{gold}\right\|_\infty.\end{aligned}$$

□

The next lemma is the generalized linear model analog of our earlier Lemma 2, and relies on an argument made in Theorem 1 of Chen et al. (1999).

LEMMA 8. *On the event $\mathcal{I}$, we have that both*

$$\|\mathbf{X}_{gold}\nu\|_2^2 \leq \frac{dn_{gold}}{m^2\psi^2 n_{proxy}^2}\lambda_1\,, \qquad \textit{and} \qquad \|\nu\|_1 \leq \frac{\sqrt{d\lambda_1}}{m\psi n_{proxy}}\,.$$

Proof of Lemma 8 Recall the generalized linear model's maximum likelihood equation

$$\hat{\beta}_{proxy} = \arg\min_{\beta} \sum_{i=1}^{n_{proxy}} \left\{-Y_{proxy}^{(i)}\beta^\top \mathbf{X}_{proxy}^{(i)} + A\left(\beta^\top \mathbf{X}_{proxy}^{(i)}\right) - B\left(Y_{proxy}^{(i)}\right)\right\}.$$

The first-order optimality condition for $\hat{\beta}_{proxy}$ yields

$$\sum_{i=1}^{n_{proxy}} \mathbf{X}_{proxy}^{(i)}\left(Y_{proxy}^{(i)} - A'\left(\hat{\beta}_{proxy}^\top \mathbf{X}_{proxy}^{(i)}\right)\right) = 0\,.$$

Substituting $Y_{proxy}^{(i)} = \mu\left({\beta_{proxy}^*}^\top \mathbf{X}_{proxy}^{(i)}\right) + \varepsilon_{proxy}^{(i)}$ and $A' = \mu$, we get

$$\sum_{i=1}^{n_{proxy}} \mathbf{X}_{proxy}^{(i)}\left(\mu\left(\hat{\beta}_{proxy}^\top \mathbf{X}_{proxy}^{(i)}\right) - \mu\left({\beta_{proxy}^*}^\top \mathbf{X}_{proxy}^{(i)}\right)\right) = \sum_{i=1}^{n_{proxy}} \mathbf{X}_{proxy}^{(i)}\varepsilon_{proxy}^{(i)} = \mathbf{X}_{proxy}^\top \varepsilon_{proxy}\,. \tag{23}$$

Now, note that by applying the mean value theorem, that there exists some β_0 on the line segment between β_{proxy}^* and $\hat{\beta}_{proxy}$ satisfying

$$\mu\left(\hat{\beta}_{proxy}^\top \mathbf{X}_{proxy}^{(i)}\right) - \mu\left({\beta_{proxy}^*}^\top \mathbf{X}_{proxy}^{(i)}\right) = \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot\left(\hat{\beta}_{proxy} - \beta_{proxy}^*\right)^\top \mathbf{X}_{proxy}^{(i)}\,.$$

Recall that $\nu = \hat{\beta}_{proxy} - \beta_{proxy}^*$. Substituting the previous expression into Eq. (23) and employing a trick from Chen et al. (1999), we can write

$$\begin{aligned}
\left\|\left(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy}\right)^{-1}\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2^2 &= \left\|\left(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy}\right)^{-1}\sum_{i=1}^{n_{proxy}} \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\left(\hat{\beta}_{proxy} - \beta_{proxy}^*\right)\right\|_2^2 \\
&= \nu^\top\left(\sum_{i=1}^{n_{proxy}} \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)\left(\sum_{i=1}^{n_{proxy}} \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)^{-2} \\
&\quad\times\left(\sum_{i=1}^{n_{proxy}} \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)\nu \\
&\geq \nu^\top\left(\sum_{i=1}^{n_{proxy}} \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)\left(\sum_{i=1}^{n_{proxy}} \frac{\mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)}{m}\mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)^{-2} \\
&\quad\times\left(\sum_{i=1}^{n_{proxy}} \mu'\left(\beta_0^\top \mathbf{X}_{proxy}^{(i)}\right)\cdot \mathbf{X}_{proxy}^{(i)}{\mathbf{X}_{proxy}^{(i)}}^\top\right)\nu \\
&= m^2\|\nu\|_2^2\,,
\end{aligned}$$

where we have used the fact that $\mu'(z) \geq m > 0$ for all z in the domain since A is strongly convex (see Lemma 16). Thus, we have

$$\|\nu\|_2 \leq \frac{1}{m}\left\|\left(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy}\right)^{-1}\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2.$$

We now proceed exactly as we did in Lemma 2 in the linear case; we omit several algebraic details to avoid repetition. For the first inequality, on event $\mathcal{I}$, we can write

$$\begin{aligned}
\|\mathbf{X}_{gold}\nu\|_2^2 &\leq \frac{1}{m^2}\left\|\mathbf{X}_{gold}(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1}\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2^2 \\
&\leq \frac{1}{m^2}\|\mathbf{X}_{gold}\|_2^2\cdot\left\|(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1}\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2^2 \\
&\leq \frac{dn_{gold}}{m^2\psi^2 n_{proxy}^2}\lambda_1\,.
\end{aligned}$$

For the second inequality, we observe that on event $\mathcal{I}$,

$$\begin{aligned}\|\nu\|_1 &\leq \sqrt{d}\,\|\nu\|_2 \\ &\leq \frac{\sqrt{d}}{m}\left\|(\mathbf{X}_{proxy}^\top \mathbf{X}_{proxy})^{-1}\right\|_2 \cdot \left\|\mathbf{X}_{proxy}^\top \varepsilon_{proxy}\right\|_2 \\ &\leq \frac{\sqrt{d\lambda_1}}{m\psi n_{proxy}}\,.\end{aligned}$$

□

C.2. Proof of Lemma 9

LEMMA 9. *On the event $\mathcal{J}$, taking $\lambda \geq 5\lambda_0/2$, the solution $\hat{\delta}$ to the optimization problem* (18) *satisfies*

$$\lambda\left\|\tilde{\delta}-\hat{\delta}\right\|_1 \;\leq\; \frac{5m}{8n_{gold}}\left\|\mathbf{X}_{gold}\nu\right\|_2^2 + \frac{5\lambda^2 s}{2m\phi^2} + 5\lambda\left\|\nu\right\|_1\,.$$

Proof of Lemma 9 Since the optimization problem (18) is convex, it recovers the in-sample global minimum. Thus, we must have that

$$\mathcal{L}(\hat{\delta};\; \mathbf{X}_{gold}, Y_{gold}) + \lambda\left\|\hat{\delta}\right\|_1 \;\leq\; \mathcal{L}(\tilde{\delta};\; \mathbf{X}_{gold}, Y_{gold}) + \lambda\left\|\tilde{\delta}\right\|_1\,.$$

We can re-write this as

$$\begin{aligned}\mathcal{E}(\hat{\delta}) + \lambda\left\|\hat{\delta}\right\|_1 &\leq -\left[w(\hat{\delta}) - w(\tilde{\delta})\right] + \lambda\left\|\tilde{\delta}\right\|_1 \\ &\leq \frac{1}{n_{gold}}\left\|\hat{\delta}-\tilde{\delta}\right\|_1 \cdot \left\|\varepsilon_{gold}^\top \mathbf{X}_{gold}\right\|_\infty + \lambda\left\|\tilde{\delta}\right\|_1 \\ &\leq \frac{\lambda}{5}\left\|\hat{\delta}-\tilde{\delta}\right\|_1 + \lambda\left\|\tilde{\delta}\right\|_1 \\ &= \frac{\lambda}{5}\left\|\hat{\delta}-\delta^*+\nu\right\|_1 + \lambda\left\|\delta^*-\nu\right\|_1\,, \qquad (24)\end{aligned}$$

where we have used Lemma 7 in the second inequality, the fact that $\mathcal{J}$ holds and that $\lambda \geq 5\lambda_0/2$ in the third inequality, and the fact that $\nu = \delta^* - \tilde{\delta}$ in the last equality. Now, similar to the linear case (see Lemma 1), we use the triangle inequality to express $\hat{\delta}$ in terms of its components on the index set S. Collecting Eqs. (9)–(10) and substituting into Eq. (24), we obtain

$$5\mathcal{E}(\hat{\delta}) + 5\lambda\left(\|\delta_S^*\|_1 - \left\|\hat{\delta}_S - \delta_S^*\right\|_1 + \left\|\hat{\delta}_{S^c}\right\|_1\right) \;\leq\; \lambda\left(\left\|\hat{\delta}_S - \delta_S^*\right\|_1 + \left\|\hat{\delta}_{S^c}\right\|_1 + \|\nu\|_1\right) + 5\lambda\left(\|\delta^*\|_1 + \|\nu\|_1\right)\,.$$

Cancelling terms on both sides yields

$$5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 \;\leq\; 6\lambda\left(\left\|\hat{\delta}_S - \delta_S^*\right\|_1 + \|\nu\|_1\right)\,. \qquad (25)$$

As in the linear case, we have two possible cases: either (i) $\|\nu\|_1 \leq \left\|\hat{\delta}_S - \delta_S^*\right\|_1$, or (ii) $\left\|\hat{\delta}_S - \delta_S^*\right\|_1 < \|\nu\|_1$. In Case (i), we will invoke the compatibility condition to prove our finite-sample guarantee for the joint estimator, and in Case (ii), we will find that we already have good control over the error of the estimator.

Case (i): We are in the case that $\|\nu\|_1 \leq \left\|\hat{\delta}_S - \delta_S^*\right\|_1$, so from Eq. (25), we can write on $\mathcal{J}$,

$$5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 \;\leq\; 12\lambda\left\|\hat{\delta}_S - \delta_S^*\right\|_1\,.$$

Dropping the first (non-negative) term on the left hand side, we immediately observe that on $\mathcal{J}$,

$$\left\|\hat{\delta}_{S^c}\right\|_1 = \left\|\hat{\delta}_{S^c} - \delta_{S^c}^*\right\|_1 \;\leq\; 3\left\|\hat{\delta}_S - \delta_S^*\right\|_1\,,$$

so we can apply the compatibility condition (Definition 2) to $u = \hat{\delta} - \delta^*$ and take the square root. This yields

$$\left\|\hat{\delta}_S - \delta^*_S\right\|_1 \;\leq\; \frac{\sqrt{s}}{\phi\sqrt{n_{gold}}} \left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2 .$$

Separately, when Case (i) and $\mathcal{J}$ hold, we can further simplify

$$\begin{aligned} 5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\tilde{\delta} - \hat{\delta}\right\|_1 &= 5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\hat{\delta} - \delta^* + \nu\right\|_1 \\ &\leq 5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\hat{\delta}_S - \delta^*_S\right\|_1 + 4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 + 4\lambda\left\|\nu\right\|_1 \\ &\leq 10\lambda\left\|\hat{\delta}_S - \delta^*\right\|_1 + 10\lambda\left\|\nu\right\|_1 \\ &\leq \frac{10\lambda\sqrt{s}}{\phi\sqrt{n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2 + 10\lambda\left\|\nu\right\|_1 . \end{aligned} \tag{26}$$

where we used Eq. (25) in the second inequality, and the compatibility condition to bound $\left\|\hat{\delta}_S - \delta^*\right\|_1$ in the third inequality.

We now need to relate the expected error relative to the true minimizer $\mathcal{E}(\hat{\delta})$ to $\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2$ in order to (partially) cancel the term $\frac{10\lambda\sqrt{s}}{\phi\sqrt{n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2$ on the right hand side. In the linear case, these quantities are trivially related since $\mathcal{E}(\delta) = \frac{1}{n_{gold}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2^2$. When considering more general nonlinear loss functions, one needs to additionally impose a margin condition and a suitable alternative compatibility condition in order to establish a relationship between these two terms (see, e.g., Negahban et al. 2009, Bühlmann and Van De Geer 2011). However, this additional infrastructure is not necessary for generalized linear models due to their close connection to linear models. Lemma 6 shows that it is sufficient to use the functional form of the GLM log likelihood and the strong convexity of $A(\cdot)$ to establish a relationship.

Applying Lemma 6, we can write

$$\begin{aligned} \frac{5m}{2n_{gold}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \tilde{\delta}\right)\right\|_2^2 + 4\lambda\left\|\tilde{\delta} - \hat{\delta}\right\|_1 &\leq 5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\tilde{\delta} - \hat{\delta}\right\|_1 \\ &\leq \frac{10\lambda\sqrt{s}}{\phi\sqrt{n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2 + 10\lambda\left\|\nu\right\|_1 \\ &\leq \frac{5m}{2n_{gold}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2^2 + \frac{10\lambda^2 s}{m\phi^2} + 10\lambda\left\|\nu\right\|_1 \\ &\leq \frac{5m}{2n_{gold}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \tilde{\delta}\right)\right\|_2^2 + \frac{5m}{2n_{gold}}\left\|\mathbf{X}_{gold}\nu\right\|_2^2 + \frac{10\lambda^2 s}{m\phi^2} + 10\lambda\left\|\nu\right\|_1 , \end{aligned}$$

where we have used Eq. (26) in the second inequality, and the identity that $2ab \leq a^2 + b^2$ for

$$a = \sqrt{\frac{5m}{2n_{gold}}}\left\|\mathbf{X}_{gold}\left(\hat{\delta} - \delta^*\right)\right\|_2 \qquad \text{and} \qquad b = \frac{\lambda}{\phi}\sqrt{\frac{10s}{m}},$$

in the third inequality. Cancelling terms on both sides, we find that when $\mathcal{J}$ and Case (i) hold, we have that

$$\lambda\left\|\tilde{\delta} - \hat{\delta}\right\|_1 \;\leq\; \frac{5m}{8n_{gold}}\left\|\mathbf{X}_{gold}\nu\right\|_2^2 + \frac{5\lambda^2 s}{2m\phi^2} + \frac{5\lambda\left\|\nu\right\|_1}{2} . \tag{27}$$

Case (ii): We are in the case that $\left\|\hat{\delta}_S - \delta^*_S\right\|_1 \leq \left\|\nu\right\|_1$, so Eq. (25) implies on $\mathcal{J}$,

$$5\mathcal{E}(\hat{\delta}) + 4\lambda\left\|\hat{\delta}_{S^c}\right\|_1 \;\leq\; 12\lambda\left\|\nu\right\|_1 .$$

In this case, we do not actually need to invoke the compatibility condition. When $\mathcal{J}$ and Case (ii) hold, we can directly bound

$$\begin{aligned} 5\mathcal{E}(\hat{\delta}) + 4\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 &\leq 5\mathcal{E}(\hat{\delta}) + 4\lambda \left\| \hat{\delta}_S - \delta_S^* \right\|_1 + 4\lambda \left\| \hat{\delta}_{S^c} \right\|_1 + 4\lambda \left\| \nu \right\|_1 \\ &\leq 20\lambda \left\| \nu \right\|_1 , \end{aligned}$$

where we used the triangle inequality (see Eq. (10) from the proof in the linear case) in the first inequality and Eq. (14) as well as the fact that $\left\| \hat{\delta}_S - \delta_S^* \right\|_1 \leq \left\| \nu \right\|_1$ in the second inequality. Dropping the first (non-negative) term on the left hand side yields

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \leq 5\lambda \left\| \nu \right\|_1 . \tag{28}$$

Combining the inequalities from Eq. (27) and (28), the following holds in both cases on $\mathcal{J}$,

$$\lambda \left\| \tilde{\delta} - \hat{\delta} \right\|_1 \leq \frac{5m}{8n_{gold}} \left\| \mathbf{X}_{gold} \nu \right\|_2^2 + \frac{5\lambda^2 s}{2m\phi^2} + 5\lambda \left\| \nu \right\|_1 . \tag{29}$$

□

C.3. Proof of Theorem 6 and Corollary 2

Having established Lemma 9, the proof of this theorem follows the proof in the linear case closely. Note that we have defined the same high probability events $\mathcal{J}$ and $\mathcal{I}$ for the generalized linear model setting, and thus these events satisfy the same tail inequalities, allowing us to re-use Lemmas 4 and 5.

Proof of Theorem 6 By Lemma 9, we can bound $\left\| \tilde{\delta} - \hat{\delta} \right\|_1$ with high probability on $\mathcal{J}$ when ν is small. We can re-use Lemma 2 from the linear case to bound the terms that depend on ν as a function of n_{proxy} on the event $\mathcal{I}$. Applying Lemma 8 to Lemma 9 yields

$$\left\| \tilde{\delta} - \hat{\delta} \right\|_1 \leq \frac{5d\lambda_1}{8m\psi^2 n_{proxy}^2 \lambda} + \frac{5\lambda s}{2m\phi^2} + \frac{5\sqrt{d\lambda_1}}{m\psi n_{proxy}} .$$

The above expression holds with probability 1 when the events $\mathcal{J}$ and $\mathcal{I}$ hold, and $\lambda \geq 5\lambda_0/2$. Recall that λ_0, λ_1 are theoretical quantities that we can choose freely to optimize our bound. In contrast, λ is a fixed regularization parameter chosen by the decision-maker when training the estimator. Then, setting $\lambda_0 = 2\lambda/5$, we can write

$$\begin{aligned} \Pr\left[\left\| \tilde{\delta} - \hat{\delta} \right\|_1 \geq \frac{5d\lambda_1}{8m\psi^2 n_{proxy}^2 \lambda} + \frac{5\lambda s}{2m\phi^2} + \frac{5\sqrt{d\lambda_1}}{m\psi n_{proxy}} \right] &\leq 1 - \Pr[\mathcal{J} \cap \mathcal{I}] \\ &\leq \Pr[\mathcal{J}^C] + \Pr[\mathcal{I}^C] \\ &\leq 2d \exp\left(-\frac{\lambda^2 n_{gold}}{50\sigma_{gold}^2} \right) + 2d \exp\left(-\frac{\lambda_1}{2d\sigma_{proxy}^2 n_{proxy}} \right) . \end{aligned}$$

The second inequality follows from a union bound, and the third follows from Lemma 4 (setting $\lambda_0 = 2\lambda/5$) and Lemma 5. By inspection, we choose

$$\lambda_1 = \frac{n_{proxy}^2 \lambda^2}{d} ,$$

yielding

$$\Pr\left[\left\| \tilde{\delta} - \hat{\delta} \right\|_1 \geq \frac{5\lambda}{m} \left(\frac{1}{8\psi^2} + \frac{1}{\psi} + \frac{s}{2\phi^2} \right) \right] \leq 2d \exp\left(-\frac{\lambda^2 n_{gold}}{50\sigma_{gold}^2} \right) + 2d \exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2 \sigma_{proxy}^2} \right) .$$

Finally, we reverse our variable transformation by substituting $\hat{\beta}_{joint} = \hat{\delta} + \hat{\beta}_{proxy}$ and $\beta_{gold}^* = \tilde{\delta} + \hat{\beta}_{proxy}$, which gives us the result. □

The following corollary uses the tail inequality in Theorem 6 to obtain an upper bound on the expected parameter estimation error of the truncated joint estimator.

COROLLARY 2 **(Joint Estimator)**. *The parameter estimation error of the truncated nonlinear joint estimator for a generalized linear model is bounded above as follows:*

$$R\left(\hat{\beta}^{tr}_{joint}(\lambda), \beta^*_{gold}\right) \;\leq\; \frac{5\lambda}{m}\left(\frac{1}{8\psi^2}+\frac{1}{\psi}+\frac{s}{2\phi^2}\right)+6bd\left(\exp\left(-\frac{\lambda^2 n_{gold}}{50\sigma^2_{gold}}\right)+\exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma^2_{proxy}}\right)\right).$$

Let $C>0$ be any tuning constant. Taking the regularization parameter to be

$$\bar{\lambda} \;=\; \max\left\{\sqrt{\frac{50\sigma^2_{gold}\log\left(6bdn_{gold}\right)}{n_{gold}}},\; \sqrt{\frac{2d^2\sigma^2_{proxy}\log\left(6bdn_{proxy}\right)}{n_{proxy}}}\right\} = \;\tilde{\mathcal{O}}\left(\frac{\sigma_{gold}}{\sqrt{n_{gold}}}+\frac{d\sigma_{proxy}}{\sqrt{n_{proxy}}}\right),$$

yields a parameter estimation error of order

$$R\left(\hat{\beta}^{tr}_{joint}(\bar{\lambda}), \beta^*_{gold}\right) \;=\; \tilde{\mathcal{O}}\left(\frac{s\sigma_{gold}}{\sqrt{n_{gold}}}+\frac{sd\sigma_{proxy}}{\sqrt{n_{proxy}}}\right).$$

Proof of Corollary 2 For ease of notation, let

$$w = \frac{5}{m}\left(\frac{1}{8\psi^2}+\frac{1}{\psi}+\frac{s}{2\phi^2}\right).$$

Recall from the proof of Theorem 6 that

$$\left\|\hat{\beta}_{joint}(\lambda)-\beta^*_{gold}\right\|_1 \;\leq\; \lambda w\,,$$

with probability 1 when the events $\mathcal{J}$ and $\mathcal{I}$ hold, and we take $\lambda_0 = 2\lambda/5$ and $\lambda_1 = n^2_{proxy}\lambda^2/d$.

From here, we can use the proof of Corollary 1 exactly, and thus avoid repeating the steps. Collecting Eqs. (21) and (22) and substituting into Eq. (20) from the proof in the linear case, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\beta}^{tr}_{joint}(\lambda)-\beta^*_{gold}\right\|_1\right] \;&\leq\; \lambda w\cdot\Pr[\mathcal{J}\cap\mathcal{I}]+3b\cdot\Pr[\mathcal{J}^C\cup\mathcal{I}^C]\\
&\leq \lambda w+3b\cdot\left(\Pr[J^C]+\Pr[\mathcal{I}^C]\right)\\
&\leq \lambda w+6bd\cdot\left(\exp\left(-\frac{\lambda^2 n_{gold}}{50\sigma^2_{gold}}\right)+\exp\left(-\frac{\lambda^2 n_{proxy}}{2d^2\sigma^2_{proxy}}\right)\right),
\end{aligned}$$

using a union bound, and applying Lemmas 4 and 5 with the chosen values $\lambda_0 = 2\lambda/5$ and $\lambda_1 = n^2_{proxy}\lambda^2/d$.

□

Appendix D: Useful Lemmas

D.1. Properties of Gaussians

LEMMA 10. *Consider a zero-mean multivariate gaussian random variable, $z\sim\mathcal{N}(0,\Sigma)$. Then,*

$$\mathbb{E}\left[\|z\|_1\right] \;\geq\; \sqrt{\frac{2}{\pi}}\,\mathrm{tr}\left(\Sigma^{1/2}\right).$$

Proof of Lemma 10 We begin with the case where z is a scalar ($d=1$) and $\Sigma=\sigma^2$. Then, we can write

$$\begin{aligned}
\mathbb{E}[|z|] &= \int_{-\infty}^{\infty}|z'|\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{z'^2}{2\sigma^2}}dz'\\
&= 2\int_0^{\infty}|z'|\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{z'^2}{2\sigma^2}}dz'\\
&= \sqrt{\frac{2\sigma^2}{\pi}}\int_0^{\infty}e^{-u}du\\
&= \sqrt{\frac{2\sigma^2}{\pi}},
\end{aligned}$$

where we have used a variable substitution $u = \frac{z'^2}{2\sigma^2}$ (implying $du = \frac{z'}{\sigma^2} dz'$).

Then, for the case where z is a vector ($d \geq 1$),

$$\begin{aligned}
\mathbb{E}\left[\|z\|_1\right] &= \sum_{i=1}^{d} \mathbb{E}[|z_i|] \\
&\geq \sum_{i=1}^{d} \sqrt{\frac{2\Sigma_{ii}}{\pi}} \\
&= \sqrt{\frac{2}{\pi}} \operatorname{tr}\left(\Sigma^{1/2}\right) .
\end{aligned}$$

□

LEMMA 11. *Consider a multivariate gaussian random variable with mean μ, $z \sim \mathcal{N}(\mu, \Sigma)$. Then,*

$$\mathbb{E}\left[\|z\|_1\right] \;\geq\; \max\left\{\frac{1}{2}\|\mu\|_1, \frac{1}{\sqrt{2\pi}} \operatorname{tr}\left(\Sigma^{1/2}\right)\right\} .$$

Proof of Lemma 11 We begin with the case where z is a scalar ($d = 1$), so $\Sigma = \sigma^2$. Without loss of generality, assume $\mu \geq 0$; if not, we can equivalently consider $-z$ instead of z, since $|-z| = |z|$ and $-z \sim \mathcal{N}(-\mu, \sigma^2)$. Next, observe that $|z + \mu| \geq |z|$ if $z \geq 0$, so we can write

$$\begin{aligned}
\mathbb{E}[|z|] &= \int_{-\infty}^{\infty} |z' + \mu| \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&\geq \int_{0}^{\infty} |z' + \mu| \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&\geq \int_{0}^{\infty} z' \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&= \sqrt{\frac{\sigma^2}{2\pi}} \int_{0}^{\infty} e^{-u} du \\
&= \sqrt{\frac{\sigma^2}{2\pi}} ,
\end{aligned}$$

where we have used a variable substitution $u = \frac{z'^2}{2\sigma^2}$ (implying $du = \frac{z'}{\sigma^2} dz'$).

In addition, observe that $|z + \mu| \geq |\mu|$ if $z \geq 0$, so we can write

$$\begin{aligned}
\mathbb{E}[|z|] &= \int_{-\infty}^{\infty} |z' + \mu| \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&\geq \int_{0}^{\infty} |z' + \mu| \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&\geq |\mu| \int_{0}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z'^2}{2\sigma^2}} dz' \\
&= \frac{1}{2}|\mu| .
\end{aligned}$$

Then, for the case where z is a vector so $d \geq 1$, we can write

$$\begin{aligned}
\mathbb{E}\left[\|z\|_1\right] &= \sum_{i=1}^{d} \mathbb{E}[|z_i|] \\
&\geq \sum_{i=1}^{d} \max\left\{\frac{1}{2}|\mu_i|, \sqrt{\frac{\Sigma_{ii}}{2\pi}}\right\} \\
&\geq \max\left\{\sum_{i=1}^{d} \frac{1}{2}|\mu_i|, \sum_{i=1}^{d} \sqrt{\frac{\Sigma_{ii}}{2\pi}}\right\} \\
&= \max\left\{\frac{1}{2}\|\mu\|_1, \frac{1}{\sqrt{2\pi}} \operatorname{tr}\left(\Sigma^{1/2}\right)\right\} .
\end{aligned}$$

□

LEMMA 12. *Consider a multivariate gaussian random variable* $x \sim \mathcal{N}(\mu, \Sigma)$. *Then,*

$$\mathbb{E}\left[\|z\|_2^2\right] = \|\mu\|_2^2 + \operatorname{tr}(\Sigma) .$$

Proof of Lemma 12

$$\begin{aligned}
\mathbb{E}\left[\|z\|_2^2\right] &= \mathbb{E}\left[\|\mu + (z-\mu)\|_2^2\right] \\
&= \|\mu\|_2^2 + \mathbb{E}\left[(z-\mu)^\top (z-\mu)\right] \\
&= \|\mu\|_2^2 + \operatorname{tr}\left(\mathbb{E}\left[(z-\mu)(z-\mu)^\top\right]\right) \\
&= \|\mu\|_2^2 + \operatorname{tr}(\Sigma) .
\end{aligned}$$

□

D.2. Properties of Subgaussians

LEMMA 13 **(Concentration Inequality for Subgaussians)**. *Let* z *be a* σ*-subgaussian random variable (see Definition 1). Then, for all* $t \geq 0$,

$$\Pr[|z| \ \geq \ t] \ \leq \ 2\exp\left(-\frac{t^2}{2\sigma^2}\right) .$$

Proof of Lemma 13 See Eq. (2.9) of Wainwright (2016). □

LEMMA 14 **(Hoeffding Bound for Subgaussians)**. *Let* $\{z_i\}_{i=1}^n$ *be a set of independent* σ*-subgaussian random variables (see Definition 1). Then, for all* $t \geq 0$,

$$\Pr\left[\sum_{i=1}^n z_i \ \geq \ t\right] \ \leq \ \exp\left(-\frac{t^2}{2n\sigma^2}\right) .$$

Proof of Lemma 14 See Proposition 2.1 of Wainwright (2016). □

LEMMA 15. *Let* $\{z_i\}_{i=1}^n$ *be a set of independent* σ*-subgaussian random variables, and let* $\{a_i\}_{i=1}^n$ *be constants that satisfy* $A = \sum_{i=1}^n a_i^2$. *Then,*

$$W = \sum_{i=1}^n a_i z_i$$

is a $\left(\sigma\sqrt{A}\right)$*-subgaussian random variable as well.*

Proof of Lemma 15

$$\begin{aligned}
\mathbb{E}[\exp(tW)] &= \mathbb{E}\left[\exp\left(t\sum_{i=1}^n a_i z_i\right)\right] \\
&= \prod_{i=1}^n \mathbb{E}[\exp(t a_i z_i)] \\
&\leq \prod_{i=1}^n \exp\left(\frac{\sigma^2 t^2 a_i^2}{2}\right) \\
&= \exp\left(\frac{\sigma^2 t^2}{2}\sum_{i=1}^n a_i^2\right) \\
&= \exp\left(\sigma^2 t^2 A/2\right) ,
\end{aligned}$$

implying that W is $\left(\sigma\sqrt{A}\right)$-subgaussian by Definition 1. □

D.3. Strong Convexity

The following lemma is a simplification of a more general result in Nesterov (1998), and shows that a uniform lower bound on the second derivative of a real-valued function implies strong convexity.

LEMMA 16. *Consider a twice-differentiable function $f:\mathbb{R}\to\mathbb{R}$ with a uniform lower bound on its second derivative for all points in its domain $\mathcal{X}\subset\mathbb{R}$, i.e.,*

$$\inf_{x\in\mathcal{X}} f''(x)\geq m\,,$$

Then, for all $x,y\in\mathcal{X}$, f satisfies

$$f(y)-f(x)\ \geq\ f'(x)\cdot(y-x)+\frac{m}{2}\cdot(y-x)^2\,.$$

Proof of Lemma 16 Note that

$$f(y)-f(x)=\int_x^y f'(t)dt \qquad \text{and} \qquad f'(t)=f'(x)+\int_x^t f''(s)ds\,.$$

Then, we can write

$$\begin{aligned} f(y)-f(x) &= \int_x^y \left(f'(x)+\int_x^t f''(s)ds\right)dt \\ &= f'(x)(y-x)+\int_x^y\int_x^t f''(s)dsdt \\ &\geq f'(x)(y-x)+m\cdot\int_x^y\int_x^t dsdt \\ &= f'(x)(y-x)+\frac{m}{2}\cdot(y-x)^2\,. \end{aligned}$$

□

Appendix E: Experimental Details & Results

E.1. Additional Simulations

We now simulate several additional benchmarks on synthetic data.

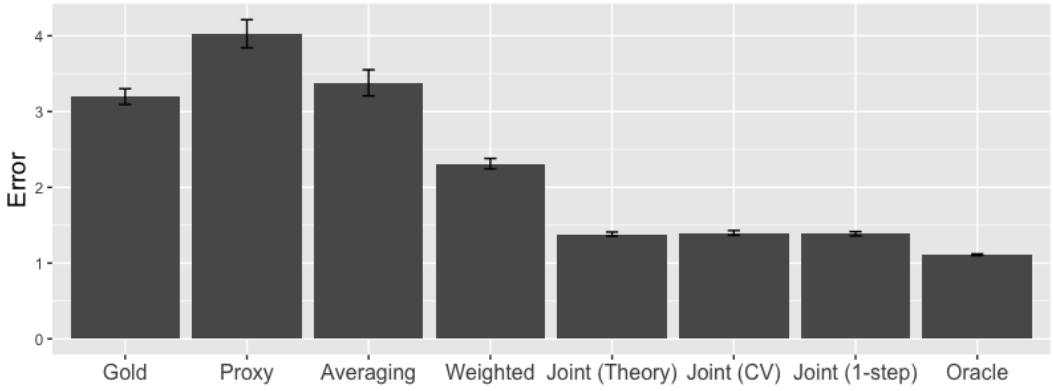
1. *High dimension:* The setting in Section 5.1 is relatively low-dimensional since $n_{gold}>d$. We investigate the high-dimensional setting, where $n_{gold}<d<n_{proxy}$, by taking $d=200, n_{gold}=150$ and $n_{proxy}=n_{test}=2000$. All other parameters and data-generating processes remain the same as in Section 5.1.

2. *Theoretically-prescribed regularization parameter:* Corollary 1 suggests taking the regularization parameter (in the linear case) to be $\bar{\lambda}$, where

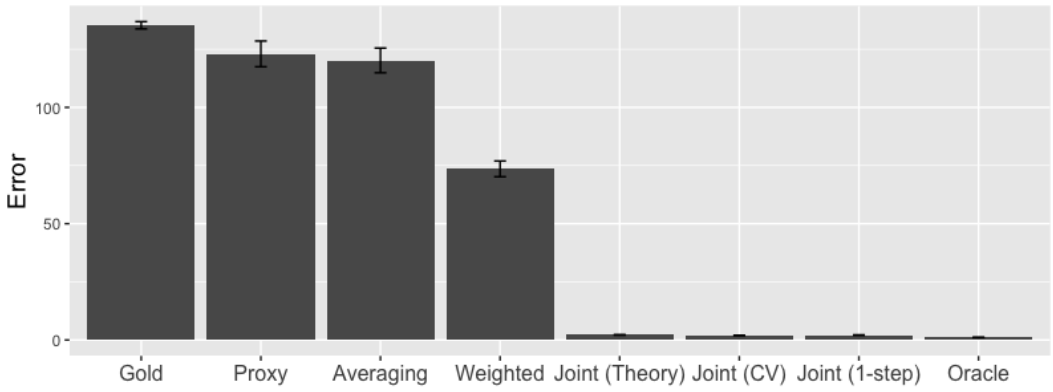
$$\bar{\lambda}\ =\ C\max\left\{\sqrt{\frac{\sigma_{gold}^2\log\left(6bdn_{gold}\right)}{n_{gold}}},\ \sqrt{\frac{2d^2\sigma_{proxy}^2\log\left(6bdn_{proxy}\right)}{n_{proxy}}}\right\}.$$

We tune the multiplicative constant C *once* on a validation set (based on data generated as described in Section 5.1) and set $C=1/10$; unlike cross-validation, this choice of regularization parameter is now held fixed across all randomly-generated datasets (i.e., iterations) and parameter values (i.e., when the values of n_{gold}, n_{proxy}, d change in the high-dimensional setting). Since we are using synthetically generated data, we know the appropriate values of $\sigma_{gold}, \sigma_{proxy}$ and b. We investigate the performance of this choice of regularization parameter compared to its cross-validated counterpart using the glmnet package[8] in R.

[8] To match the normalization in the implementation of glmnet, we must divide our regularization parameter by n_{gold}.



(a) Low Dimension



(b) High Dimension

Figure 5 Out-of-sample prediction error and 95% confidence intervals of different estimators on synthetic data.

3. *One-step joint estimator:* In Section 4.6, we described a "one-step" version of the joint estimator that simultaneously estimates the proxy and gold parameters, while enforcing an ℓ_1 penalty on the difference of the two parameters. We implemented this estimator using the CVXR package in R, and tuned the regularization parameter via cross-validation on a validation set.

Figure 5a shows results for the low-dimensional setting (described in Section 5.1) and Figure 5b shows results for the high-dimensional setting (described above). First, we note that the performance gain from using the joint estimator rather than the baseline estimators (gold, proxy, averaging and weighted) is far larger in high dimension compared to low dimension. This is to be expected, since it becomes more important to leverage sparse structure as d grows large. Second, in both low and high dimension, we find that the performance of the joint estimator with the theoretically-prescribed regularization parameter [denoted "Joint (Theory)"] closely matches the performance of the joint estimator with the cross-validated regularization parameter [denoted "Joint (CV)"]. This empirically validates the form of the regularization parameter proposed in Corollary 1. However, in practice, one does not know problem-specific values $(\sigma_{gold}, \sigma_{proxy}, b)$ needed to compute $\bar{\lambda}$, making cross-validation attractive. Third, in both low and high dimension, we find little to no improvement from combining the estimation of $\hat{\beta}_{proxy}$ and $\hat{\beta}_{gold}$ using the one-step joint estimator [denoted "Joint (1-step)"]. As we argued in Section 4.6, we expect this to be the case when $n_{proxy} \gg n_{gold}$ (as we have here), since the one-step estimation decouples into our two-step joint estimator.

E.2. Expedia Case Study Details

The original dataset has 9,917,530 impressions. A subset of this data includes impressions where the hotels were randomly sorted, i.e., when the provided feature random_bool = 1. As recommended, we restrict ourselves to this subset to avoid the position bias of the existing algorithm. This results in 2,939,652 impressions.

There are 54 columns in the data. We drop the following columns:

1. Features that are missing more than 25% of their entries
2. Unique identifiers for the search query, property, customer, country of property, country of customer, and search query destination
3. Time of search
4. Boolean used to identify the subset of impressions that were randomly sorted
5. Number of rooms and nights, after being used to normalize the overall price per room-night

There are 15 remaining features and 2 outcome variables (clicks and bookings). These include: the property star rating (1-5), average customer review rating for the property (1-5), an indicator for whether the hotel is part of a major hotel chain, two different scores outlining the desirability of the hotel's location, the logarithm of the mean price of the hotel over the last trading period, the hotel position on Expedia's search results page, the displayed price in USD of the hotel for the given search, an indicator whether the hotel had a displayed sale price promotion, the length of stay, the number of days in the future the hotel stay started from the search date, the number of adults, the number of children, the number of rooms, and an indicator for a weekend stay. We also drop impressions that have missing values and outliers at the 99.99% level, leaving 2,262,166 total impressions. As recommended by Friedman et al. (2001), we standardize each feature before performing any regressions.

As described in Section 5.2, we use a subsample of 10,000 randomly drawn observations in each iteration. We first reserve 50% of this data as a held-out test set to assess performance. The remaining 50% is used as the training set for the gold and proxy estimators. For the averaging, weighted, and joint estimators, we additionally need to choose a tuning parameter. Following standard practice, we use a random 70% subsample of the observations (that are not in the test set) as our training set and the remaining 30% as a validation set. We train models with different values of λ on the training set, and use mean squared prediction error on the validation set to choose the best value of λ in the final model. Finally, the "Oracle" is trained on all 2M+ impressions excluding the test set.

E.3. Diabetes Case Study Details

The original dataset has 9948 patient records across 379 healthcare providers. The data only contains patients who have recently visited the provider at least twice. Each patient is associated with 184 features constructed from patient-specific information available *before* the most recent visit (i.e., indicator variables for past ICD-9 diagnoses, medication prescriptions, and procedures), as well as a binary outcome variable *from* the most recent visit (i.e., whether s/he was diagnosed with diabetes in the last visit). As described in Section 5.3, we only study 3 of the 379 providers: we use patient data from a medium-sized provider as our gold data, and patient data pooled from two larger providers as our proxy data.

However, we use patient data from the remaining 376 providers to do variable selection as a pre-processing step. In particular, we run a simple LASSO variable selection procedure by regressing diabetes outcomes against the 184 total features (note that we exclude the 3 healthcare providers that constitute the proxy and gold populations in this step to avoid overfitting). This leaves us with roughly 100 commonly predictive features (depending on the randomness in the cross-validation procedure).

We first reserve 50% of the gold data as a held-out test set to assess performance, and use the remaining 50% of the gold data for training models. Unlike the Expedia case study, our test set does not overlap with the proxy data (since the gold and proxy data are derived from different cohorts) so the entire proxy data is used for training. The gold estimator is trained on all gold observations that are not in the test set; the proxy estimator is trained on all proxy observations. For the averaging, weighted, and joint estimators, we additionally need to choose a tuning parameter. Following standard practice (modified to accommodate proxy data), we use a random 70% subsample of the gold observations (that are not in the test set) and all of the proxy observations as our training set; we use the remaining 30% of the gold observations as our validation set. We train models with different values of λ on the training set, and use mean squared prediction error on the validation set to choose the best value of λ in the final model.