Statistica Sinica **34** (2024), 2277-2300
doi:https://doi.org/10.5705/ss.202022.0242

DISTRIBUTED LOGISTIC REGRESSION FOR MASSIVE DATA WITH RARE EVENTS

Xuetong Li[1], Xuening Zhu*[2] and Hansheng Wang[1]

[1]*Peking University and* [2]*Fudan University*

Abstract: Large-scale rare events data are commonly encountered in practice. To tackle the massive rare events data, we propose a novel distributed estimation method for logistic regression in a distributed system. A distributed framework faces the following two challenges. The first challenge is how to distribute the data. Here, we investigate two distribution strategies, namely, the RANDOM strategy and the COPY strategy. The second challenge is how to select an appropriate type of objective function so that the best asymptotic efficiency can be achieved. Then, the under-sampled (US) and inverse probability weighted (IPW) types of objective functions are considered. Our results suggest that the COPY strategy with the IPW objective function is the best solution for a distributed logistic regression with rare events. We demonstrate the finite sample performance of the distributed methods using simulation studies and a real-world Swedish Traffic Sign dataset.

Key words and phrases: Distributed system, logistic regression, massive rare events data.

1. Introduction

Massive data with rare events in binary regression are commonly encountered in scientific fields and applications. Conceptually, rare events data, also called imbalanced data, refer to the number of instances in the positive class being much smaller than that in the negative class. For example, in online search or recommendation systems, billions of impressions can be generated each day. If we treat each impression as one sample, then the probability for one impression to generate a click is very small. Thus, we can treat clicks as rare events (Japkowicz (2000); McMahan et al. (2013); Chen et al. (2016); Huang et al. (2020)). As another example in political science, the occurrence of wars, vetos, coups and the decisions of citizens to run for office have been modeled as rare events (King and Zeng (2001); Owen (2007); Neunhoeffer and Sternberg (2019)). Our last example is small object detection in a high resolution image; see Figure 1. Suppose we treat each pixel as a sample and whether it is covered by a bounding box as corresponding response. Then, the bounding box of a small object treated as a positive instance only covers less than 1% of the original image (Zhu et al. (2016); Zhao et al. (2019); Chen et al. (2022)). Other important rare events data

*Corresponding author.

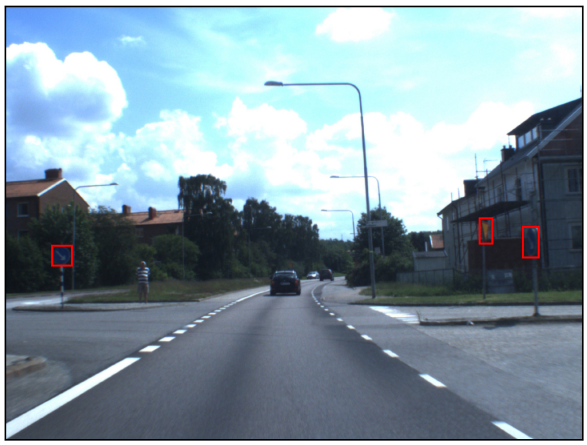

Figure 1. An example in the Swedish Traffic Sign dataset for traffic sign detection. The original image is of size $960 \times 1,280 \times 3$. Each bounding box is used to annotate a local region containing a traffic sign. The bounding box of a small object treated as a positive instance only covers less than 1% of the original image.

examples include fraud detection (Bolton and Hand (2002); Hassan and Abraham (2016)), drug discovery (Zhu, Su and Chipman (2006); Korkmaz (2020)) and rare disease diagnosis (Zhuang et al. (2019)). For a comprehensive summary, we refer to Sun, Wong and Kamel (2009), Haixiang et al. (2017) and Kaur, Pannu and Malhi (2019).

A common approach to tackle imbalanced data is to balance it by undersampling the negative class (Drummond and Holte (2003); Liu, Wu and Zhou (2008); Nguyen, Cooper and Kamei (2012)) or oversampling the positive class (Chawla et al. (2002); Han, Wang and Mao (2005); Mathew et al. (2017)). Most existing literature focuses on practical algorithms and methodologies for classification with few statistical theory guarantees. They design sampling strategies or ensemble learning methods to improve classification accuracy (Krawczyk (2016)). For example, Estabrooks, Jo and Japkowicz (2004) empirically investigated an effective combination of different resampling paradigms to improve classification accuracy. Sun et al. (2007) adapted the AdaBoost algorithm for advancing the classification of imbalanced data. King and Zeng (2001) considered logistic regression in rare events data and focused on correcting the biases when estimating the regression coefficients and probabilities. Fithian and Hastie (2014) used the special structure of logistic regression models to design a novel local case-control sampling method. However, these theoretical studies are based on the regular assumption that the probability of event occurring is fixed. This might not be the best way to describe rare events mathematically, because this assumption implies that the number of rare events should diverge to infinity at the same rate as the total sample size diverges towards infinity. Instead, for rare events, it is

more appropriate to assume that the positive class rate should decay towards zero as the total sample size increases.

In this regard, Wang (2020) developed a novel theoretical framework and the resulting estimators' statistical properties were investigated accordingly. Under his novel theoretical framework, he showed that the convergence rate of the global maximum likelihood estimator (GMLE) is mainly determined by the number of positive instances instead of the total sample size. As a consequence, the convergence rate of the GMLE should be considerably slower than that of the usual cases. Additionally, Wang (2020) surprisingly found that both under-sampling and over-sampling methods would cause unnecessary statistical efficiency loss in parameter estimation. Then, how to develop new estimation methods so that a statistically efficient estimator can be obtained becomes a problem of great importance. In the remainder of the paper, we call an estimator to be statistically efficient, if it achieves the same asymptotic distribution as the GMLE.

It is worth mentioning that we are not among the first group of researchers studying the problem of logistic regression for massive data. Significant progresses have been made in the past literature. One possible solution is subsampling. For example, Wang, Zhu and Ma (2018) developed a subsampling method motivated by the A-optimality criterion of Kiefer (1959). Wang (2019) further proposed more efficient estimators based on subsamples with optimal subsampling probabilities. A general model with imbalanced binary response is studied by Wang, Zhang and Wang (2021) recently. Another possible solution is distributed computing, if a parallel computing system can be used. For example, Du, Li and Li (2018) proposed differentially private approaches to collaboratively and accurately train a logistic regression model among multiple parties. Shi, Wang and Zhang (2019) studied a distributed logistic regression based on the classical ADMM algorithm (Boyd et al. (2011)). Zuo et al. (2021) proposed a distributed subsampling procedure to approximate the maximum likelihood estimator. A cost-sensitive algorithm was developed by Wang et al. (2016) for the linear support vector machine problem. Despite the usefulness of the above methods, few attempts have been made for distributed classification problems with rare events data and rigorous asymptotic theory. Without a solid theoretical foundation, we are not able to deliver a statistically efficient estimator. This motivates us to develop a novel distributed logistic regression method with solid statistical theory support for massive rare events data.

It is noteworthy that developing a distributed estimation method for logistic regression with rare events is not straightforward. We face at least the following two challenging problems. The first problem is data distribution on local computers in a distributed system. Because the total number of positive instances is much smaller than the total sample size, the traditional pure random data distribution strategy might not be the best choice in some cases. For example, if

the number of instances assigned to a local machine is very small, this traditional strategy leads to even smaller positive instances for each distributed computer node. This process makes the local estimates obtained from each local computer statistically inaccurate, which in turn makes the finally combined estimator statistically inefficient. In fact, a potentially better choice is to copy all the positive instances to each local computer and then to distribute the negative instances to local computers as randomly as possible. For convenience, we refer to the traditional data distribution strategy as a fully RANDOM strategy and this new strategy as a COPY strategy. Then, investigating the statistical properties of the estimators under both RANDOM and COPY strategies becomes a problem of great interest.

The second problem is the choice of objective function. If the COPY strategy is adopted, the positive and negative instances become much more balanced on each local computer, which makes the statistical estimation easier. However, the side effect is that the local objective function is no longer unbiased for the global log-likelihood function. Thus, the resulting estimator is statistically inefficient, even though the resulting estimator remains to be asymptotically normal. This is an interesting finding of Wang (2020). For convenience, we refer to the estimator computed on each local computer as an under-sampled estimator. To solve this problem, a new-type objective function is proposed on each local computer, which should be unbiased for the global one. This naturally leads to an inverse probability weighted estimator (Fithian and Hastie (2014); Wang (2020)). Subsequently, we consider obtaining a distributed logistic regression estimator. A simple and common approach is to take the average of the estimators produced by the local computers. This approach is referred to the one-shot (OS) method in the literature (Zhang, Duchi and Wainwright (2013); Rosenblatt and Nadler (2016); Chang, Lin and Wang (2017)). We use the OS method to combine the local IPW estimators to yield the final estimator, which is referred to as the IPW estimator.

To summarize, we aim to make the following important contributions to the existing literature. First, we theoretically prove that the traditional RANDOM distributed framework cannot perform efficiently with rare events data due to its unignorable random bias term in many cases. Second, a COPY strategy is proposed and rigorously investigated. The US type of local objective function is used to construct a US estimator. We find that the US estimator has a lower bias but unsatisfactory statistical efficiency if the number of negative instances on each computer node is not enough. Lastly, we find that the IPW estimator is statistically more efficient than the US estimator and has the same asymptotic behavior as the GMLE. Theoretical findings are further verified by extensive numerical studies.

The remainder of this paper is organized as follows. Section 2 introduces the model setting and three important benchmark estimation methods according to

Wang (2020). Section 3 presents three distributed estimation methods and their asymptotic theory. Numerical studies are given in Section 4. An application to the Swedish traffic Sign Data is illustrated here using these three distributed methods. The article concludes with a brief discussion in Section 5. All technical details are relegated to the online Supplementary Material.

2. Logistic Regression with Rare Events Data

2.1. Model setup

Suppose there are N observations in total, which are indexed by $1 \le i \le N$. The ith observation is denoted as (X_i, Y_i), where $X_i \in \mathbb{R}^p$ is a p-dimensional covariate and $Y_i \in \{0, 1\}$ is the binary response. Assume (X_i, Y_i) is independently generated for $1 \le i \le N$ and denote the full data by $\mathcal{S}_F = \{(X_i, Y_i) : 1 \le i \le N\}$. Let $N_1 = \sum_{i=1}^{N} Y_i$ be the number of positive instances, and $N_0 = N - N_1$ be the number of negative instances. To model their regression relationship, the following logistic regression model is considered

$$P(Y_i = 1 \mid X_i) = p_i(\alpha, \beta) = \frac{e^{\alpha + X_i^\top \beta}}{1 + e^{\alpha + X_i^\top \beta}}, \tag{2.1}$$

where $\alpha \in \mathbb{R}$ is the intercept and $\beta \in \mathbb{R}^p$ is the slope parameter. Define $\theta = (\alpha, \beta^\top)^\top \in \mathbb{R}^{p+1}$ as the full parameter vector with true value given by $\theta^* = (\alpha^*, \beta^{*\top})^\top$. As N diverges to infinity, if θ^* does not change, the number of positive instances would diverge at a rate of $O_p(N)$. Following Shao (2003), we define $O_p(\cdot)$ as follows. Let $\{A_i\}$ and $\{B_i\}$ with $1 \le i \le N$ be two random variable sequences. We then say $A_i = O_p(B_i)$ if and only if for any $\varepsilon > 0$ there is a constant $C_\varepsilon > 0$, such that $\sup_i P(\|A_i\| \ge C_\varepsilon \|B_i\|) < \varepsilon$.

Under the classical logistic regression model setting (2.1), existing theory shows that the maximum likelihood estimator (MLE) based on the full data $\mathcal{S}_F$ converges at a rate of $O_p(N^{-1/2})$ (Nelder and Wedderburn (1972)). As convincingly argued by Wang (2020), this might not be the best choice for modeling rare events data. For rare events data, the percentage of positive instances is extremely small. Statistically, it is more appropriate to specify the positive response rate to converge towards 0 as the total sample size increases towards infinity. Meanwhile, we wish the covariate effect (as measured by β^*) remains constant since the value of X is unknown. Otherwise, it cannot be accurately estimated statistically. Consequently, this suggests that we should replace the intercept parameter α^* by α_N^*, which should diverge towards negative infinity as $N \to \infty$. Specifically, we should have $\alpha_N^* \to -\infty$ at an appropriate divergence rate as $N \to \infty$. However, what is a reasonable divergence rate requires more careful investigation. Under this assumption, we should have $P(Y_i = 1 \mid X_i) \approx e^{\alpha_N^* + X_i^\top \beta^*}$ as $N \to \infty$. We then have $E(N_1) \approx N e^{\alpha_N^*} E(e^{X_i^\top \beta^*})$.

Even though the positive response rate (i.e., N_1/N) should converge toward zero as N goes to infinity, we still expect that the total number of positive instances (i.e., N_1) should diverge to infinity. Otherwise, we cannot estimate the parameters of interest consistently. This suggests we should have

$$\alpha_N^* \to -\infty \quad \text{and} \quad \alpha_N^* + \log N \to \infty \tag{2.2}$$

when $N \to \infty$. This becomes the most important technical assumption for the proposed theoretical framework (Wang (2020)).

2.2. Related methods

In this subsection, we demonstrate a number of important benchmark estimation methods according to Wang (2020). Specifically, we introduce the global maximum likelihood estimation, under-sampled estimation, and inverse probability weighted likelihood estimation, respectively.

2.2.1. Global maximum likelihood estimation

We start with the global maximum likelihood estimation method using the full data. The log-likelihood function based on the full data $\mathcal{S}_F$ is given as follows:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \Big[Y_i \log p_i(\alpha_N, \beta) + (1 - Y_i) \log \{1 - p_i(\alpha_N, \beta)\} \Big], \tag{2.3}$$

where $p_i(\alpha_N, \beta) = e^{\alpha_N + X_i^\top \beta}/(1 + e^{\alpha_N + X_i^\top \beta})$. Then we obtain the GMLE as $\widehat{\theta}_{\text{GMLE}} = \text{argmax}_\theta \mathcal{L}(\theta)$. According to Theorem 1 in Wang (2020), the GMLE $\widehat{\theta}_{\text{GMLE}}$ should be $\sqrt{N e^{\alpha_N^*}}$-consistent and asymptotically normal under appropriate conditions. This result suggests that the convergence rate of the GMLE is fully determined by the number of positive instances, which implies that the help provided by an extra large amount of the negative instances should be limited. This result is particularly true when the total number of negative instances is too large to be easily managed on one computer.

Nevertheless, we should remark that this never implies that a large number of negative instances is totally useless for efficiency improvement. Extensive theoretical and numerical experiences suggest that the statistical efficiency of various benchmark estimators can be improved by a more efficient use of negative instances, even though the convergence rate remains unchanged (Wang (2020)). However, for many practical datasets with rare events, the total number of negative instances is often too large to be easily managed on one computer. In this case, how to utilize negative instances more efficiently for better estimation efficiency becomes a problem of great interest.

2.2.2. Under-sampled estimation

In practice, researchers often seek to include all the positive instances for statistical analysis, because they are rare and thus valuable (Drummond and Holte (2003); Liu, Wu and Zhou (2008); Nguyen, Cooper and Kamei (2012)). Next, the same (or comparable) number of negative instances are randomly selected so that a more balanced subsample can be constructed. Subsequently, interested parameters can be estimated based on this more balanced subsample. For convenience, we refer to this common practice as an under-sampled method (Drummond and Holte (2003); Liu, Wu and Zhou (2008); Nguyen, Cooper and Kamei (2012); Wang (2020)). By doing so, the estimation problem becomes computationally feasible. Theoretically, this problem can be formulated as follows. Let a_i be a binary indicator with $P(a_i = 1) = \pi$, which is independently generated for each i. Here, $a_i = 1$ suggests that the ith instance is sampled and π is the probability for sampling. Accordingly, the US objective function becomes

$$\mathcal{L}_{\mathrm{US}}(\theta) = \sum_{i=1}^{N} \Big[Y_i \log p_i(\alpha_N, \beta) + (1 - Y_i) a_i \log \{1 - p_i(\alpha_N, \beta)\} \Big]. \tag{2.4}$$

For convenience, we call it a US objective function. Then, we obtain a US estimator as $\widehat{\theta}_{\mathrm{US}} = \mathrm{argmax}_\theta \mathcal{L}_{\mathrm{US}}(\theta)$. However, Wang (2020) finds that $\widehat{\theta}_{\mathrm{US}}$ is a biased estimator for θ^*. Thus, the debiased US estimator is further obtained as $\widetilde{\theta}_{\mathrm{US}} = \widehat{\theta}_{\mathrm{US}} + (\log \pi, 0, \ldots, 0)^\top$.

Comparing (2.4) with (2.3), we find the only difference is the treatment of the negative instances. Considering (2.3), all the instances are used regardless of positives or negatives. However, in (2.4), we use all positive instances, and include negative instances only if the corresponding binary indicator $a_i = 1$. By doing so, we include all positive instances and only a much smaller number of negative instances. One can verify easily that this formulation is mathematically equivalent to that of Wang (2020). The careful theoretical analysis of Wang (2020) suggests that such an estimator remains to be $\sqrt{Ne^{\alpha_N^*}}$-consistent and is asymptotically normal. However, as shown in Theorem 3 by Wang (2020), the US estimator cannot obtain the same efficiency as that of the GMLE if the ratio of positive instances to negative instances does not converge to zero.

2.2.3. Inverse probability weighted estimation

The key reason for the statistical inefficiency of the US estimator is the objective function in (2.4). By under-sampling, the resulting objective function has been materially changed. A direct consequence is that it is no longer an unbiased estimator for the global log-likelihood function. That leads to the inefficiency for the US estimator. To fix this problem, one possible solution is to find an unbiased estimator for the global log-likelihood function. This leads to the following objective function for inverse probability weighted estimation

(King and Zeng (2001); Fithian and Hastie (2014); Wang (2020))

$$\mathcal{L}_{\mathrm{IPW}}(\theta) = \sum_{i=1}^{N} \left[Y_i \log p_i(\alpha_N, \beta) + \frac{(1 - Y_i) a_i \log\{1 - p_i(\alpha_N, \beta)\}}{\pi} \right]. \tag{2.5}$$

One can easily verify that $E\{\mathcal{L}_{\mathrm{IPW}}(\theta)|\mathcal{S}_F\} = \mathcal{L}(\theta)$, which suggests that $\mathcal{L}_{\mathrm{IPW}}(\theta)$ is an unbiased estimator for the global log-likelihood function. By optimizing the above objective function, an IPW estimator can be obtained as $\widehat{\theta}_{\mathrm{IPW}} = \mathrm{argmax}_\theta \mathcal{L}_{\mathrm{IPW}}(\theta)$. Wang (2020) demonstrated that the IPW-type estimator has the same convergence rate $O_p(1/\sqrt{Ne^{\alpha_N^*}})$ as that of $\widehat{\theta}_{\mathrm{GMLE}}$ but remains to be statistically inefficient. Recall that we define in this work an estimator to be statistically efficient if it shares the same asymptotic distribution as the GMLE.

The suboptimal efficiency of both the US and IPW estimators is understandable because both methods include only a very small fraction of the negative instances for estimation. Then, there should exist a good possibility to use a larger number of negative instances (but not as large as the full set of negative class) for better statistical efficiency. This seems to be a particularly promising direction if a powerful distributed computing system is available. With the help of a distributed system, we should be able to compute various local estimators (e.g., the US and IPW estimators) multiple times. They can then be aggregated together to form a more powerful estimator. However, what type of local estimators should be computed and how they should be assembled so that the final estimator can be as efficient as the GMLE are problems of great interest. We thus aim to systematically investigate these interesting problems in the next sections.

3. Distributed Logistic Regression

3.1. Distributed MLE with random strategy

We start with the simplest distributed estimator, that is the distributed MLE obtained under the RANDOM strategy. For convenience, we refer to this as RMLE. Assume there exists a distributed computation system with a total of K local computers and one central computer. A typical architecture of a distributed system is shown in Figure 2. The local computers are indexed by $1 \le k \le K$. Then, the RMLE method randomly distributes the full data $\mathcal{S}_F$ to each local computer with approximately equal sizes. Denote $\mathcal{S}_F = \mathcal{S}_+ \cup \mathcal{S}_-$, where $\mathcal{S}_+ = \{i : Y_i = 1\}$ represents the set of all the positive instances, and $\mathcal{S}_- = \{i : Y_i = 0\}$ represents the set of all negative instances. Specifically, let $\mathcal{S}_k^R$ be the sample randomly distributed to the kth local computer with $\mathcal{S}_k^R = \mathcal{S}_{k+}^R \cup \mathcal{S}_{k-}^R$, where $\mathcal{S}_{k+}^R = \{i : i \in \mathcal{S}_k^R, Y_i = 1\}$ and $\mathcal{S}_{k-}^R = \{i : i \in \mathcal{S}_k^R, Y_i = 0\}$ refer to the set of positive and negative instances on the kth local computer, respectively. For convenience, denote $n_k = |\mathcal{S}_k^R|$. In addition, let $n_{1k}^R = |\mathcal{S}_{k+}^R|$ and $n_{0k}^R = |\mathcal{S}_{k-}^R|$. Mathematically, denote $a_i^{(k)} = 1$ if the ith observation is randomly distributed

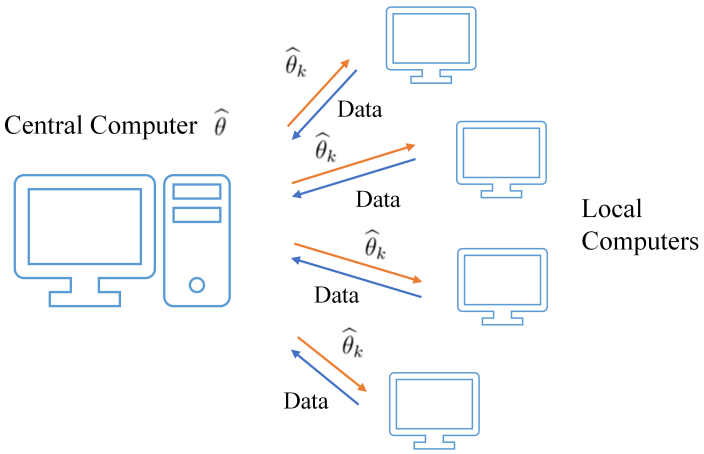

Figure 2. Illustration of the distributed system.

to kth local computer. We then have $\sum_{k=1}^{K} a_i^{(k)} = 1$ for every i, $n_k = \sum_{i=1}^{N} a_i^{(k)}$. We also define $n = E(n_k) = N/K$. Additionally, we have $n_{1k} = \sum_{i=1}^{N} a_i^{(k)} Y_i$ and $n_{0k} = \sum_{i=1}^{N} a_i^{(k)}(1 - Y_i)$.

As one can see, by the RANDOM strategy, both the positive and negative instances are randomly distributed to each local computer. As a consequence, their relative percentages remain approximately the same as the full data size. That is $n_{1k}/n_k \approx N_1/N$. The merit of this method is that the data distribution on each local computer remains the same as that of the full data. However, the drawback is that the number of positive instances allocated to each local computer become even smaller. This might turn into statistical inefficiency for the resulting estimator. Specifically, for each local computer, define

$$\mathcal{L}_{\mathrm{R},k}(\theta) = \sum_{i=1}^{N} a_i^{(k)} \Big[Y_i \log p_i(\alpha_N, \beta) + (1 - Y_i) \log\{1 - p_i(\alpha_N, \beta)\} \Big]$$

as a local log-likelihood function with $P(a_i^{(k)} = 1) = 1/K$. Then a local MLE is computed as $\widehat{\theta}_{\mathrm{RMLE},k} = \operatorname{argmax}_\theta \mathcal{L}_{\mathrm{R},k}(\theta)$. Then, each local computer should report this local estimator to the central computer. Next, the central computer assembles those estimators to form a more powerful estimator. To achieve this goal, a typical assembling solution is the OS type strategy (Zhang, Duchi and Wainwright (2013); Chang, Lin and Wang (2017)). More specifically, the final estimator is given by $\widehat{\theta}_{\mathrm{RMLE}} = \sum_{k=1}^{K} \widehat{\theta}_{\mathrm{RMLE},k}/K$. The asymptotic distribution of $\widehat{\theta}_{\mathrm{RMLE}}$ is presented in the following theorem.

Theorem 1. *Assume* (C1) $P(\|Z_i\| > M) \le 2\exp(-C_{\mathrm{Tail}} M^2)$ *with* $Z_i = (1, X_i^\top)^\top \in \mathbb{R}^{p+1}$ *for some positive constant* C_{Tail}, (C2) $n \to \infty$ *as* $N \to \infty$, (C3) $\log^2 N/(ne^{\alpha_N^*}) = O(1)$ *and* (2.2). *Then we have the following asymptotic*

representation

$$\sqrt{Ne^{\alpha_N^*}}\big(\widehat{\theta}_{\mathrm{RMLE}}-\theta^*\big)=P_1+\frac{P_2}{2}+o_p\bigg(\frac{K}{\sqrt{Ne^{\alpha_N^*}}}\bigg),$$

where $P_1=-(Ne^{\alpha_N^*})^{1/2}K^{-1}\sum_{k=1}^{K}\ddot{\mathcal{L}}_{\mathrm{R},k}^{-1}(\theta^*)\dot{\mathcal{L}}_{\mathrm{R},k}(\theta^*)$ *and* $P_2=(Ne^{\alpha_N^*})^{-1/2}K$ $B(\theta^*)$. *Here* $B(\theta^*)$ *is a random bias term such that* $C_{\min}\le E\{\|B(\theta^*)\|\}\le C_{\max}$ *for some fixed positive constants* $0<C_{\min}<C_{\max}<\infty$.

The theorem condition (C1) requires that covariate distributions have an exponentially decayed tail probability (Zhang and Chen (2020)). The theorem condition (C2) implies that the expected number of the instances on each local computer $n=E(n_k)$ should diverge to infinity as the total sample size $N\to\infty$. In the meanwhile, we have $E(n_{1k})=E(\sum_{i=1}^{N}a_i^{(k)}Y_i)=ne^{\alpha_N^*}E\{e^{X_i^\top\beta^*}/(1+e^{Z_i^\top\theta^*})\}$. By condition (C3), we require that the number of the positive instances on the local computer should be large enough. Then by Theorem 1, we know that $\sqrt{Ne^{\alpha_N^*}}(\widehat{\theta}_{\mathrm{RMLE}}-\theta^*)$ can be decomposed into three parts. The first part is P_1, where $P_1\to_d N(0,\Sigma^{*-1})$ as $N\to\infty$. The second part P_2 is a random bias term of order $K/(Ne^{\alpha_N^*})$, where the analytical formula for $B(\theta^*)$ is given in Supplementary Material S1. The third part is a higher order and negligible term as compared with P_2. If K is sufficiently small in the sense that $K/\sqrt{Ne^{\alpha_N^*}}\to 0$ as $N\to\infty$, we should have P_1 being the leading term. In this case, $\widehat{\theta}_{\mathrm{RMLE}}$ shares the same asymptotic distribution as the $\widehat{\theta}_{\mathrm{GMLE}}$ of Wang (2020). Otherwise, we should have $P_2/2$ as the dominating term. This makes the statistical efficiency of $\widehat{\theta}_{\mathrm{RMLE}}$ poor.

3.2. Under-sampling with an unweighted objective function

Next, we study the asymptotic properties of the distributed estimators by under-sampling. We start with $\widehat{\theta}_{\mathrm{US}}$ utilized by the unweighted loss function (2.4). To obtain the US estimator, we distribute the full data $\mathcal{S}_F$ to each local computer by the COPY strategy. Let $\mathcal{S}_k^C$ be the sample distributed to the kth local computer under the COPY strategy. Denote $\mathcal{S}_k^C=\mathcal{S}_{k+}^C\cup\mathcal{S}_{k-}^C$, where $\mathcal{S}_{k+}^C$ and $\mathcal{S}_{k-}^C$ refer to the positive and negative instances on the kth local computer, respectively. For the COPY strategy, we have $\mathcal{S}_{k+}^C=\mathcal{S}_+$ for $1\le k\le K$, which implies that the positive instances remain the same for all local computers. As one can see, the advantage of the COPY strategy is that the number of positive cases allocated to each local computer becomes much larger than that of the RANDOM method. The negative instances are then randomly distributed on each local computer such that $\cup_k\mathcal{S}_{k-}^C=\mathcal{S}_-$ with $\mathcal{S}_{k_1-}^C\cap\mathcal{S}_{k_2-}^C=\emptyset$ for any $k_1\ne k_2$. Let $n_{1k}^C=|\mathcal{S}_{k+}^C|$ and $n_{0k}^C=|\mathcal{S}_{k-}^C|$. We typically require that $n_{1k}^C=O_p(n_{0k}^C)$. In other words, the number of negative instances assigned to each local computer should not be much smaller than that of the positive instances, which is also the most common case in practice.

Subsequently, define a local MLE for each local computer as $\widehat{\theta}_{\mathrm{US},k} = \operatorname{argmax}_\theta \mathcal{L}_{\mathrm{US},k}(\theta)$, where we have

$$\mathcal{L}_{\mathrm{US},k}(\theta) = \sum_{i=1}^{N} \Big[Y_i \log p_i\big(\alpha_N, \beta\big) + \big(1 - Y_i\big) a_i^{(k)} \log\big\{1 - p_i(\alpha_N, \beta)\big\}\Big].$$

Recall that $a_i^{(k)} = 1$ if the ith instance is allocated to the kth local computer. As a consequence, the $\widehat{\theta}_{\mathrm{US},k}$ on each worker is equivalent to the under-sampled estimator proposed by Wang (2020). After conducting local estimation, each local computer sends the local estimator $\widehat{\theta}_{\mathrm{US},k}$ to the central computer. Similarly, by using the OS strategy, we obtain the final estimator as $\widehat{\theta}_{\mathrm{US}} = \sum_{k=1}^{K} \widehat{\theta}_{\mathrm{US},k}/K$. We next analyze the asymptotic properties of $\widehat{\theta}_{\mathrm{US}}$ in the following theorem.

Theorem 2. *Assume the same conditions in Theorem* 1. *Define* $\flat = (\log K, 0, \ldots, 0)$ *and* $\Sigma_2^* = E\{(1 + \gamma e^{X_i^\top \beta^*})^{-1} e^{X_i^\top \beta^*} Z_i Z_i^\top\}$ *with* $\gamma = \lim_{N\to\infty} K e^{\alpha_N^*} \in [0, \infty)$. *We then have the following asymptotic representation as*

$$\sqrt{N e^{\alpha_N^*}}\big(\widehat{\theta}_{\mathrm{US}} - \theta^* - \flat\big) = \big(N e^{\alpha_N^*}\big)^{-1/2} \Sigma_2^{*-1} K^{-1} \sum_{k=1}^{K} \dot{\mathcal{L}}_{\mathrm{US},k}\big(\theta^* + \flat\big) + o_p(1).$$

By Theorem 2, we know that $\sqrt{N e^{\alpha_N^*}}(\widehat{\theta}_{\mathrm{US}} - \theta^* - \flat)$ can be decomposed into two parts. For the first part, we have $\Sigma_2^{*-1} K^{-1} \sum_{k=1}^{K} \dot{\mathcal{L}}_{\mathrm{US},k}(\theta^* + \flat)(N e^{\alpha_N^*})^{-1/2} \to_d N(0, \Sigma_2^{*-1}\Sigma_1^*\Sigma_2^{*-1})$ as $N \to \infty$, where $\Sigma_1^* = E\{(1 + \gamma e^{X_i^\top \beta^*})^{-2} e^{X_i^\top \beta^*} Z_i Z_i^\top\}$. The second part is a higher order negligible term. Here the asymptotic normality can be established since $(N e^{\alpha_N^*})^{-1/2} \Sigma_2^{*-1} K^{-1} \sum_{k=1}^{K} \dot{\mathcal{L}}_{\mathrm{US},k}(\theta^* + \flat)$ can be written as the summation of a set of carefully defined independent random variables; see Theorem 2 Step 4 in Supplementary Material S2 for details. Therefore, the Lindeberg-Feller Central Limit Theorem can be readily applied. Consequently, $\widetilde{\theta}_{\mathrm{US}} = \widehat{\theta}_{\mathrm{US}} - \flat$ is $\sqrt{N e^{\alpha_N^*}}$-consistent for θ^*. Comparing this results with that of Theorem 1, we find some interesting differences. First, an additional bias correction term $\flat$ is necessarily involved for the intercept. It is mainly caused by the distortion of the data distribution in the US setting. Second, we find that the US estimator has a lower bias than that of the RMLE estimator if K is large. That is mainly because the bias of the local estimators computed by the COPY strategy is smaller than that of the RANDOM strategy.

We further comment about the constant γ occurring in both Σ_1^* and Σ_2^*. As remarked by Wang (2020), one can verify that $\gamma E(e^{X_i^\top \beta^*}) \approx N_1/(N_0/K)$ asymptotically, where N_0/K represents the number of negative instances on each local computer. Thus, $\gamma E(e^{X_i^\top \beta^*})$ asymptotically quantifies the ratio of the positive instance number to negative instance number. If $\gamma = 0$, then the number of negative instances dominates the positive ones. Therefore, we have $\Sigma_2^{*-1}\Sigma_1^*\Sigma_2^{*-1} = \Sigma^{*-1}$. This implies that the US estimator shares the same asymptotic covariance matrix as the GMLE $\widehat{\theta}_{\mathrm{GMLE}}$. If $0 < \gamma < \infty$, then the

positive and negative instances are of comparable sizes. This implies that the US estimator becomes statistically inefficient as compared with the GMLE $\widehat{\theta}_{\mathrm{GMLE}}$. This finding is also consistent with Theorem 2 in (Wang (2020)). We do not consider $\gamma = \infty$, which implies that the number of positive instances is much larger than that of the negative ones.

3.3. Under-sampling with a weighted objective function

The analysis presented in Sections 3.1 and 3.2 suggests that neither the RMLE nor the US estimator can achieve the global asymptotic efficiency. The RMLE fails because too small amount of positive instances are distributed to each local computer. The US estimator fails since the US objective function used by each local computer is not unbiased for the global one. We are then inspired to develop a new local log-likelihood function, which should be an unbiased estimator for the global one. Meanwhile, all positive instances should be used by each local machine. To this end, we propose an IPW estimator as follows. Specifically, we still distribute the full data $\mathcal{S}_F$ to each local computer by the COPY strategy. Next, we define for each local computer a local MLE as $\widehat{\theta}_{\mathrm{IPW},k} = \mathrm{argmax}_\theta \mathcal{L}_{\mathrm{IPW},k}(\theta)$, where we have

$$\mathcal{L}_{\mathrm{IPW},k}(\theta) = \sum_{i=1}^{N} \Big[Y_i \log p_i\big(\alpha_N, \beta\big) + K\big(1 - Y_i\big) a_i^{(k)} \log\big\{1 - p_i(\alpha_N, \beta)\big\}\Big].$$

Hence, on each worker, the $\widehat{\theta}_{\mathrm{IPW},k}$ can be treated as the under-sampled weighted estimator proposed by Wang (2020) as also given in (2.5). One can immediately verify that $E\{\mathcal{L}_{\mathrm{IPW},k}(\theta)|\mathcal{S}_F\} = \mathcal{L}(\theta)$, where recall that $\mathcal{S}_F = \{(X_i, Y_i) : 1 \le i \le N\}$ denotes the full data. Then, each local computer sends this local estimator $\widehat{\theta}_{\mathrm{IPW},k}$ to the central computer. Similarly, by using the OS strategy, we obtain the final estimator as $\widehat{\theta}_{\mathrm{IPW}} = \sum_{k=1}^{K} \widehat{\theta}_{\mathrm{IPW},k}/K$. As noted before, $\mathcal{L}_{\mathrm{IPW},k}(\theta)$ is now an unbiased estimator for the global log-likelihood function, and we expect $\widehat{\theta}_{\mathrm{IPW}}$ to achieve the same asymptotic efficiency as the GMLE. To this end, we analyze the asymptotic properties of $\widehat{\theta}_{\mathrm{IPW}}$ in the following theorem.

Theorem 3. *Assume the same conditions in Theorem* 1*, we then have the following asymptotic representation as*

$$\sqrt{Ne^{\alpha_N^*}}\big(\widehat{\theta}_{\mathrm{IPW}} - \theta^*\big) = \big(Ne^{\alpha_N^*}\big)^{-1/2} \Sigma^{*-1} \dot{\mathcal{L}}\big(\theta^*\big) + o_p(1).$$

By Theorem 3, we know that $\sqrt{Ne^{\alpha_N^*}}(\widehat{\theta}_{\mathrm{IPW}} - \theta^*)$ could be decomposed into two parts. For the first part, we have $(Ne^{\alpha_N^*})^{-1/2}\Sigma^{*-1}\dot{\mathcal{L}}(\theta^*) \to_d N(0, \Sigma^{*-1})$ as $N \to \infty$. The second part is of the order $o_p(1)$, which is a higher order negligible term. Consequently, $\widehat{\theta}_{\mathrm{IPW}}$ is $\sqrt{Ne^{\alpha_N^*}}$-consistent for θ^*. Comparing this result of the GMLE in Wang (2020), we find that $\widehat{\theta}_{\mathrm{IPW}}$ shares the same asymptotic distribution as the GMLE. Comparing the result of the US estimator in Theorem

2, we find that the US estimator over-weights the positive instances by using the US objective function (2.4) on the local computers. In contrast, the IPW estimator assigns equal weights to positive instances and negative instances by using the IPW objective function (2.5) on the local computers. This is the key reason why the IPW estimator performs better than the US estimator. Particularly, the γ given in Theorem 2 is not involved, which represents the asymptotic ratio of positive instances to negative instances. As a consequence, we do not require $\gamma = 0$ to attain the global efficiency as compared to the US estimator (or the under-sampled estimator in Wang (2020)). Our extensive numerical studies also illustrate better finite sample performance of $\widehat{\theta}_{\mathrm{IPW}}$. To summarize, the RMLE estimator $\widehat{\theta}_{\mathrm{RMLE}}$ with a large K suffers from significant bias. The debiased US estimator $\widetilde{\theta}_{\mathrm{US}}$ is statistically inefficient either due to its high asymptotic covariance if the number of negative instances distributed on each computer node is not enough. The IPW estimator $\widehat{\theta}_{\mathrm{IPW}}$ stands out as the most attractive estimator.

It is remarkable that both the US and IPW estimators investigated in Wang (2020) are different from their counterpart estimators studied in our work. Specifically, these two estimators in Wang (2020) are based on a subsample, which contains all positive instances but only a small fraction of negative instances. By doing so, a significant amount of computation cost can be nicely saved. In this case, Wang (2020) found that the US estimator is more efficient than the IPW estimator. However, both the US and IPW estimators studied in our work are based on the whole sample but computed in a distributed way. Therefore, for our estimators, not only all positive instances but also all negative instances are fully used. In fact, all the positive instances are repeatedly used by different local computers due to our COPY strategy. In contrast, only a small proportion of negative instances are used in Wang (2020). This makes the theoretical properties of our US and IPW estimators quite different from those of Wang (2020). This is also the key reason accounting for the performance differences between the two sets of estimators.

4. Numerical Studies

4.1. A simulation study

4.1.1. Model setup and performance measure

To demonstrate the finite sample performance of the proposed methods, a number of simulation studies are conducted in this section. A standard logistic regression model (2.1) is used to generate the full data with covariate $Z_i = (1, X_i^\top)^\top \in \mathbb{R}^5$. Here the covariates X_is are generated from $N(0, \Sigma)$ with $\Sigma = (\sigma_{ij})$ and $\sigma_{ij} = 0.2^{|i-j|}$. The total sample sizes are $N = 10^4, 10^5, 5 \times 10^5$ and 10^6. For a fixed N, we set $\alpha_N^* = -0.45 \log N$ and $\beta^* = (1, 1, 1, 1)^\top$. By doing so, we allow $P(Y = 1) \to 0$ and $E(N_1) \to \infty$ as $N \to \infty$. We next set the number of local

computers (i.e., K) in two different cases. For CASE 1, we set $K = 17, 36, 63, 81$ with the four different sample sizes, respectively. One can verify that, for the COPY strategy, the number of positive instances is approximately 1.5 times as large as that of the negative instances on each local computer. In contrast, for CASE 2, we set $K = 2, 3, 4, 5$ accordingly. By doing so, the number of negative instances assigned to each local computer should be much larger than that of positive instances for the COPY strategy. Next, two different distribution strategies (i.e. RANDOM and COPY) are considered. We then obtain three local estimators $\widehat{\theta}_{\mathrm{RMLE},k}$, $\widetilde{\theta}_{\mathrm{US},k}$, and $\widehat{\theta}_{\mathrm{IPW},k}$ for every local machine k. Here $\widetilde{\theta}_{\mathrm{US},k}$ and $\widehat{\theta}_{\mathrm{IPW},k}$ can be treated as the under-sampled estimators proposed by (Wang (2020)). This leads to the combined estimators as $\widehat{\theta}_{\mathrm{RMLE}}$, $\widetilde{\theta}_{\mathrm{US}}$, and $\widehat{\theta}_{\mathrm{IPW}}$ on the central computer. Here for the US method, we use the debiased estimator $\widetilde{\theta}_{\mathrm{US}}$ (instead of $\widehat{\theta}_{\mathrm{US}}$) as our final estimator. For comparison purpose, the GMLE $\widehat{\theta}_{\mathrm{GMLE}}$ is also calculated. For a reliable evaluation, each experiment is randomly replicated for a total of $M = 500$ times. Let $\widehat{\theta}^{(m)} = (\theta_j^{(m)} : 1 \le j \le p+1)^\top$ be one particular estimator obtained in the mth replication (e.g., $\widehat{\theta}_{\mathrm{RMLE},k}$ for $k = 1$ or $\widehat{\theta}_{\mathrm{RMLE}}$). To evaluate the estimation accuracy, we calculate the root mean square error (RMSE) as $\mathrm{RMSE} = (p+1)^{-1}\sum_{j=1}^{p+1}\{M^{-1}\sum_{m=1}^{M}(\widehat{\theta}_j^{(m)} - \theta_j^*)^2\}^{1/2}$. Then the RMSE of $\widehat{\theta}_{\mathrm{GMLE}}$ is numerically computed according to its theoretical formula. Furthermore, the absolute bias of $\widehat{\theta}$ is estimated by $\mathrm{BIAS} = (p+1)^{-1}\sum_{j=1}^{p+1}|\bar{\theta}_j - \theta_j^*|$, where $\bar{\theta}_j = M^{-1}\sum_{m=1}^{M}\widehat{\theta}_j^{(m)}$. The standard error (SE) of $\widehat{\theta}$ is estimated by $\mathrm{SE} = (p+1)^{-1}\sum_{j=1}^{p+1}\{M^{-1}\sum_{m=1}^{M}(\widehat{\theta}_j^{(m)} - \bar{\theta}_j)^2\}^{1/2}$.

4.1.2. Simulation results

The detailed results are given in Figure 3. Here we study both the local and distributed estimators. Two different cases (i.e., CASE 1 and CASE 2) regarding the number of local machines are considered. This leads to a total of four combinations that are represented in different panels. The vertical axis in Figure 3 represents the RMSE value in log-scale. The horizontal axis denotes the total sample size also in log-scale. First, the top left panel presents the results of the local estimators for CASE 1. In this case, all estimators under study are much less efficient than the GMLE in the sense that the log(RMSE) values of various estimators are much larger than that of the GMLE. This is because other estimators (i.e., $\widehat{\theta}_{\mathrm{RMLE},k}$, $\widetilde{\theta}_{\mathrm{US},k}$ and $\widehat{\theta}_{\mathrm{IPW},k}$) are local estimators. Here $\widetilde{\theta}_{\mathrm{US},k} = \widehat{\theta}_{\mathrm{US},k} - \flat$ is the debiased estimator with $\flat = (\log K, 0, \ldots, 0)$. The sample sizes used by these estimators are much smaller than that of the global estimator. Consequently, they are expected to be less efficient than the global estimator. However, among all local estimators, we find that the performance of $\widehat{\theta}_{\mathrm{RMLE},k}$ is always the worst. This is expected because the number of positive instances used by the RMLE estimator is much less than that of other local estimators. Comparatively speaking, we find that $\widetilde{\theta}_{\mathrm{US},k}$ performs better than

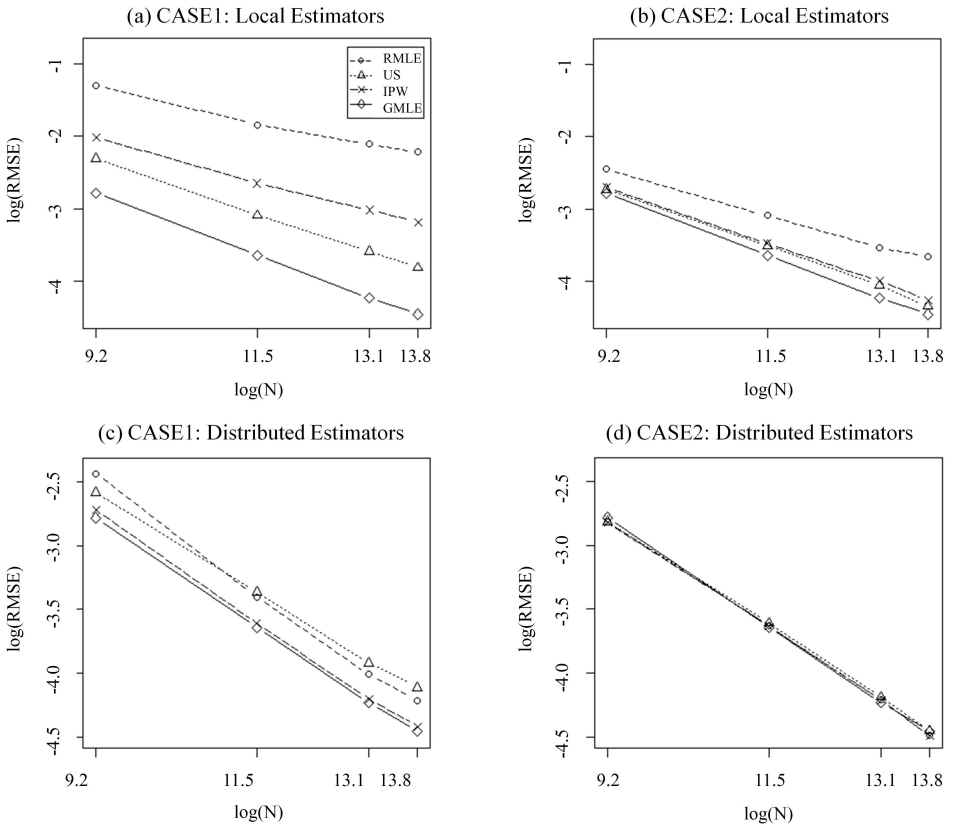

Figure 3. RMSE of the local and distributed estimators in log-scale. The horizontal axis presents the total sample size N in log-scale. The top panels show the local estimators. The bottom panels present the distributed estimators. The left panels show the cases where the number of positive cases is approximately 1.5 times as large as that of negative ones. The right panels present the cases where the number of negative instances is much larger than that of positive ones.

$\widehat{\theta}_{\mathrm{IPW},k}$. These observations are in line with that of Wang (2020).

The top right panel in Figure 3 presents the results of the local estimators for CASE 2. Compared with the top left panel, we find that the GMLE remains to be the best estimator. However, among all local estimators, the performance differences are markedly smaller. This is because the number of negative instances assigned to the local machine is sufficiently large in this case. This makes the performances of all local estimators improve towards that of the global estimator and their relative differences vanish.

The bottom left panel presents the log(RMSE) values of distributed estimators for CASE 1. We find that the performances of the distributed estimators (e.g., $\widehat{\theta}_{\mathrm{IPW}}$) are improved compared to the local estimators (e.g., $\widetilde{\theta}_{\mathrm{US,k}}$ and $\widehat{\theta}_{\mathrm{IPW,k}}$ proposed by Wang (2020)) especially when the under-sampled negative instances

Table 1. Simulation Results for the Distributed Estimators under CASE 1.

		RMLE			US			IPW		
N	$\sqrt{Ne^{\alpha_N^*}}$	BIAS	SE	RMSE	BIAS	SE	RMSE	BIAS	SE	RMSE
10^4	13	0.055	0.067	0.088	0.005	0.075	0.076	0.018	0.063	0.066
10^5	24	0.019	0.027	0.033	0.001	0.035	0.035	0.006	0.026	0.027
5×10^5	37	0.010	0.015	0.018	0.000	0.020	0.020	0.003	0.015	0.015
10^6	45	0.008	0.012	0.015	0.001	0.016	0.016	0.003	0.012	0.012

are not enough. For example, the RMSE value of $\widetilde{\theta}_{\mathrm{US,k}}$ is 0.094 and the RMSE value of $\widehat{\theta}_{\mathrm{IPW,k}}$ is 0.121 when $N = 10^4$ in the top left panel. For comparison, the RMSE value of $\widehat{\theta}_{\mathrm{IPW}}$ is 0.065, which is close to that of the GMLE (i.e., 0.061). This implies that the IPW estimator is less sensitive to the ratio of positive to negative instances. Among all distributed estimators, we find that the debiased US estimator $\widetilde{\theta}_{\mathrm{US}}$ appears to be the worst estimator in the sense that the associated log(RMSE) value is always the largest. In contrast, the IPW estimator $\widehat{\theta}_{\mathrm{IPW}}$ stands out to be the best estimator. The relative difference among different distributed estimators disappears as the number of negative instances assigned to each local machine increases. This can be seen from the results of the bottom right panel. More detailed results about Figure 3(c) are given in Table 1. By Table 1, we find that the RMLE estimator $\widehat{\theta}_{\mathrm{RMLE}}$ in CASE 1 demonstrates a large bias, since K is relatively large. In the meanwhile, the debiased US estimator $\widetilde{\theta}_{\mathrm{US}}$ suffers from high SE values. These observations are in line with the theoretical findings of the proposed Theorems 1–2.

4.2. Swedish traffic sign data analysis

4.2.1. Data processing

For illustration purpose, we present an interesting real data example. The dataset used in this study is the Swedish Traffic Sign (STS) dataset, which is publicly available at `https://www.cvl.isy.liu.se/research/datasets/traffic-signs-dataset/`. It contains a total of 1,970 annotated images with various traffic signs annotated by bounding boxes; see Figure 1 for a graphical illustration. We aim to detect the traffic signs in Figure 1 automatically. For a reliable evaluation, we randomly split the entire data into two parts. The first part contains 1,576 images (about 80% of the whole data) for training, while the remaining 394 images (about 20% of the whole data) for testing. This task contains two important steps; see Girshick et al. (2014) and Girshick (2015). For the first step, one needs to automatically detect a sufficiently tight local region containing a traffic sign from an input image without bounding box information. In the second step, one needs to classify the traffic signs detected in the local region to different categories (e.g., prohibitive, informative, warning and mandatory traffic signs). In this study, we focus on the first step.

We subsequently demonstrate how this task can be converted into a logistic regression problem, which has a large sample size and can be efficiently solved by our proposed method in a distributed way.

Specifically, each image given in the STS dataset is of relatively high resolution; see Figure 4(a). Mathematically, each image can be represented by a tensor of size $960 \times 1{,}280 \times 3$; see Figure 4(b). Next, we apply a pretrained VGG16 model on the image (Simonyan and Zisserman (2014)). The VGG16 model is a classical convolutional neural network model with a total of 13 convolutional layers. The last two fully connected layers are dropped. Then, a feature map of size $30 \times 40 \times 512$ can be extracted from the last convolutional layer; see Figure 4(c). This can be viewed as a new "image" of resolution 30×40 but with a total of 512 channels. We can then treat each pixel of this feature map as one sample. As a result, a total of $30 \times 40 = 1{,}200$ pixel samples can be generated for each image. For each pixel sample, a feature vector of 512 dimension can be constructed. Consequently, we have $p = 512$ in this case. The ith image is then denoted by $\mathbb{X}_{i,k_1,k_2} \in \mathbb{R}^{512}$ with $1 \le i \le N$, $1 \le k_1 \le 30$ and $1 \le k_2 \le 40$; see Figure 4(d). Then the total sample size is given by $N = 1{,}970 \times 1{,}200 = 2{,}364{,}000$.

We next present the details about how the response $\mathbb{Y}_{i,k_1,k_2} \in \{0,1\}$ is constructed. Define $W_i = (W_{i,k_1,k_2})$ as a binary matrix with dimensions $960 \times 1,280$ and $W_{i,k_1,k_2} \in \{0,1\}$. For a given image with the bounding box information, define $W_{i,k_1,k_2} = 1$ if the (k_1,k_2)th pixel is located in the bounding box region and $W_{i,k_1,k_2} = 0$ otherwise; see Figure 4(e) and (f). Subsequently, we partition W_i matrix into a 30×40 block matrix with equal sizes; see Figure 4(f). Specifically, we write this block matrix as $\mathbb{W}_i = (\mathbb{W}_{i,k_1,k_2})$ with $\mathbb{W}_{i,k_1,k_2} \in \mathbb{R}^{32 \times 32}$ for $1 \le k_1 \le 30$ and $1 \le k_2 \le 40$. Next, compute the average value of the block matrix $\mathbb{W}_{i,k_1,k_2}$ and denote it by μ_{i,k_1,k_2}. With the help of TensorFlow and GPU, this operation can be efficiently conducted in a fully parallel way by an average pooling operation. Define $\mathbb{Y}_{i,k_1,k_2} = I(\mu_{i,k_1,k_2} > 0.5)$. Then $\mathbb{Y}_{i,k_1,k_2}$ becomes the binary response associated with $\mathbb{X}_{i,k_1,k_2}$; see Figure 4(g) and (h). They both correspond to the same region in the original image. All data (8.59 GB, including $\mathbb{X}_{i,k_1,k_2}$ and $\mathbb{Y}_{i,k_1,k_2}$) are placed on the hard drive. A simple calculation reveals that the sample mean of $\mathbb{Y}_{i,k_1,k_2}$ is 0.225%, which is extremely small. Thus, we can treat it as the rare events data.

Since the total sample size is extremely large, we call for a distributed computation. For illustration purpose, we fix the number of local computers as $K = 50$. This leads to the sample size allocated to each local machine being approximately $N/K = 47{,}280$ by the RANDOM strategy and $N_1 + N_0/K = 52{,}491$ by the COPY strategy. Consequently, the three distributed estimators $\widehat{\theta}_{\mathrm{RMLE}}$, $\widetilde{\theta}_{\mathrm{US}}$ and $\widehat{\theta}_{\mathrm{IPW}}$ are computed based on the train data. For comparison purpose, $\widehat{\theta}_{\mathrm{GMLE}}$ is also computed by self-developed Newton-Raphson type algorithm. If this algorithm is executed on one single computer, then the time cost is extremely high. If the algorithm is executed on a distributed system,

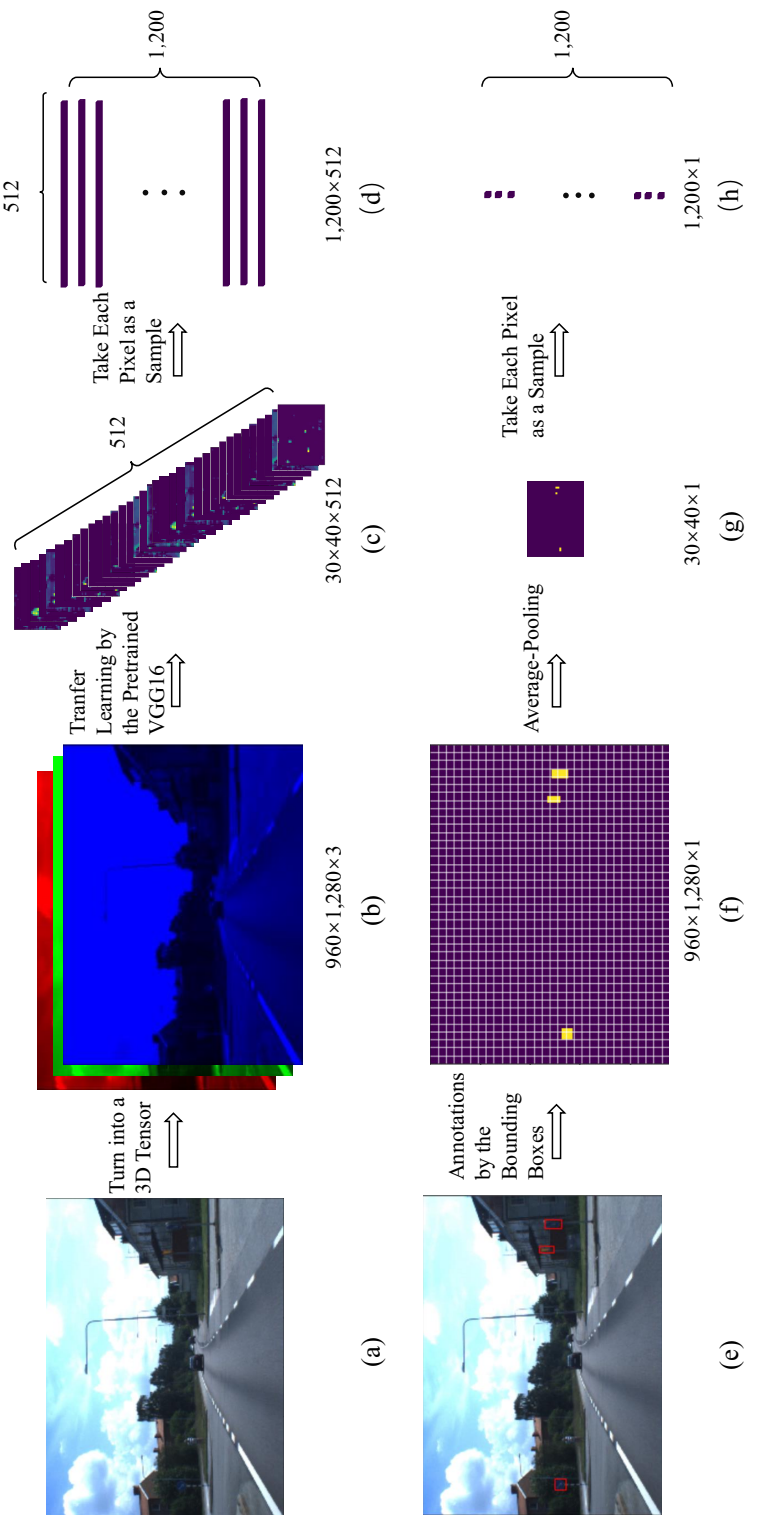

Figure 4. Illustration of the data preprocessing pipeline for one particular image. The top panel illustrates how the nonlinear features are generated. The bottom panel shows how the response is generated.

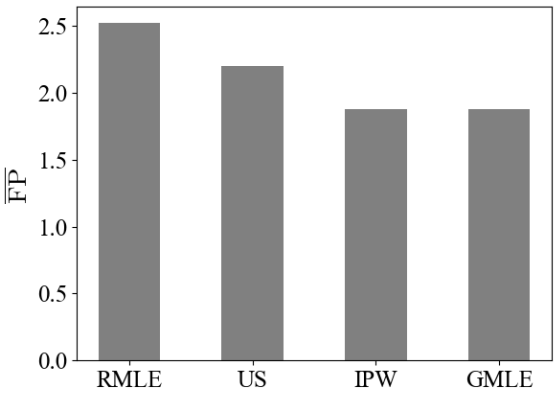

Figure 5. Prediction results $\overline{\text{FP}}$ obtained on the test data.

then the communication cost is extremely high due to the Newton-Raphson type iteration. Simply speaking, this self-developed algorithm is mainly developed here for theoretical comparison. It can hardly be used in real practice due to its high cost in time, either due to communication or computation.

4.2.2. Performance results

Next, consider the i^*th image ($1 \le i^* \le N^*$) in the test data, where $N^* = 394$ denotes the number of images for testing. For a given pixel (k_1, k_2) in the i^*th image and one particular estimator $\widehat{\theta}$ obtained on the train data (i.e., $\widehat{\theta}_{\text{RMLE}}$), we then estimate the response probability by $\widehat{p}_{i^*,k_1,k_2} = e^{\widehat{\theta}^\top \mathbb{X}_{i^*,k_1,k_2}}/(1 + e^{\widehat{\theta}^\top \mathbb{X}_{i^*,k_1,k_2}})$ and predict $\widehat{\mathbb{Y}}_{i^*,k_1,k_2} = I(\widehat{p}_{i^*,k_1,k_2} > c_{i^*})$, where $c_{i^*} = \min\{\widehat{p}_{i^*,k_1,k_2} : \mathbb{Y}_{i^*,k_1,k_2} = 1, 1 \le k_1 \le 30, 1 \le k_2 \le 40\}$. This c_i^* is the largest threshold value so that all the positive instances can be correctly captured. However, the price paid here is the false positive predictions. Define the number of the false positive instances for the i^*th image in the test data as $\text{FP}_{i^*} = \sum_{k_1,k_2} I(\widehat{\mathbb{Y}}_{i^*,k_1,k_2} = 1) I(\mathbb{Y}_{i^*,k_1,k_2} = 0)$. Its median value is then computed as FP^*. Then its overall mean across different random replications is denoted as $\overline{\text{FP}}$. The prediction results are shown in Figure 5. By Figure 5, we observe that the $\overline{\text{FP}}$ value of the IPW method is as low as 1.88, which is much smaller than 2.52 of the RMLE method and 2.20 of the US method. This value is the same as 1.88 of the GMLE method. To summarize, among all distributed estimators, the IPW estimator achieves the best performance with the smallest $\overline{\text{FP}}$ value of 1.88.

To gain further intuitive understanding about the prediction accuracy, we present several randomly selected prediction results in Figure 6. Specifically, each row in Figure 6 shows one arbitrarily selected image in the test data. The first column shows the original input image of size $960 \times 1{,}280$. The second column presents the prediction results of the US method. The third column illustrates

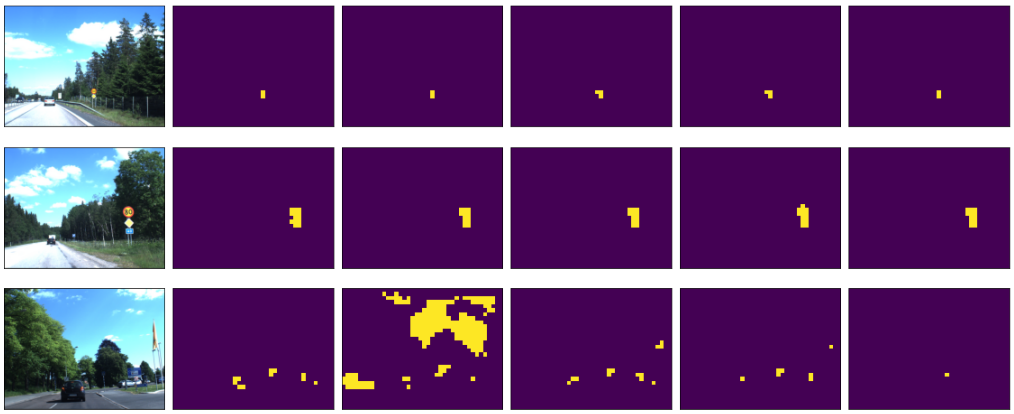

Figure 6. A number of arbitrarily selected test examples for prediction demonstration. Each row represents one arbitrarily selected image from the test data. The input images are shown in the first column. The second column presents the US method. The third column presents the RMLE method. The fourth column presents the IPW method. The fifth column presents the GMLE method. The last column shows the true annotated regions.

the prediction results by the RMLE method. The fourth column presents the prediction results due to the GMLE method. The fifth column illustrates the prediction results by the IPW method. The last column represents the true annotated regions. By Figure 6, we find that the prediction results of both US and RMLE methods are very noisy. The prediction results of both GMLE and IPW methods are much better and very comparable.

5. Conclusion

In this study, we have investigated a distributed logistic regression problem for massive rare events data. We study here two different data distribution strategies. They are RANDOM and COPY strategies, respectively. We also investigate three different estimators. They are $\widehat{\theta}_{\text{RMLE}}$, $\widetilde{\theta}_{\text{US}}$ and $\widehat{\theta}_{\text{IPW}}$, respectively. Our results suggest that the COPY strategy together with the modified log-likelihood function for the IPW estimator is the best choice. The resulting estimator can be statistically as efficient as the global estimator. To conclude this article, we would like to discuss a number of interesting topics for future study. First, we focus on the logistic regression model in this paper. It is interesting to investigate more complicated and general models in future research projects for the rare events data. Second, we use the OS strategy for the last step in the distributed estimation. Although this strategy is efficient in terms of communication, it might not be the best choice if the data are non-randomly distributed across different local machines (Zhu, Li and Wang (2021)). In this case, various inverse variance weighting (IVW) methods (Lin and Xi (2011);

Zhu, Li and Wang (2021); Yu et al. (2022)) can be used. The key idea of IVW is to take the weighted average of local estimators. The weights are related to the inverse of the Hessian matrices, which are computed by local computers. How to combine the IVW idea with our COPY strategy for distributed rare events data analysis seems to be an another interesting topic for future study. Lastly, covariates in large datasets typically have high dimensionality. Thus, how to conduct feature selection or screening based on these distributed estimators is worthy of consideration.

Supplementary Material

The online Supplementary Material contains the proofs of all theoretical results in the main text.

Acknowledgments

We thank the editor, associate editor, and two referees for their insightful comments. Xuening Zhu's research is supported by the National Natural Science Foundation of China (nos. 72222009, 71991472). Hansheng Wang's research is partially supported by National Natural Science Foundation of China (12271012, 11831008) and also partially supported by the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science (KLATASDS-MOE-ECNU-KLATASDS2101).

References

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science* **17**, 235–255.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* **3**, 1–122.

Chang, X., Lin, S.-B. and Wang, Y. (2017). Divide and conquer local average regression. *Electronic Journal of Statistics* **11**, 1326–1350.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**, 321–357.

Chen, G., Wang, H., Chen, K., Li, Z., Song, Z., Liu, Y. et al. (2022). A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52**, 936–953.

Chen, J., Sun, B., Li, H., Lu, H. and Hua, X.-S. (2016). Deep CTR prediction in display advertising. In *Proceedings of the 24th ACM International Conference on Multimedia*, 811–820.

Drummond, C. and Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II*, 1–8. Citeseer.

Du, W., Li, A. and Li, Q. (2018). Privacy-preserving multiparty learning for logistic regression. In *International Conference on Security and Privacy in Communication Systems*, 549–568. Springer.

Estabrooks, A., Jo, T. and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence* **20**, 18–36.

Fithian, W. and Hastie, T. (2014). Local case-control sampling: Efficient subsampling in imbalanced data sets. *The Annals of Statistics* **42**, 1693–1724.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H. and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* **73**, 220–239.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, 878–887. Springer.

Hassan, A. K. I. and Abraham, A. (2016). Modeling insurance fraud detection using imbalanced data classification. In *Advances in Nature and Biologically Inspired Computing*, 117–127. Springer.

Huang, J.-T., Sharma, A., Sun, S., Xia, L., Zhang, D., Pronin, P. et al. (2020). Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2553–2561.

Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*, 10–15. AAAI Press, Menlo Park.

Kaur, H., Pannu, H. S. and Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)* **52**, 1–36.

Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)* **21**, 272–304.

King, G. and Zeng, L. (2001). Logistic regression in rare events data. *Political Analysis* **9**, 137–163.

Korkmaz, S. (2020). Deep learning-based imbalanced data classification for drug discovery. *Journal of Chemical Information and Modeling* **60**, 4180–4190.

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence* **5**, 221–232.

Lin, N. and Xi, R. (2011). Aggregated estimating equation estimation. *Statistics and its Interface* **4**, 73–83.

Liu, X.-Y., Wu, J. and Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**, 539–550.

Mathew, J., Pang, C. K., Luo, M. and Leong, W. H. (2017). Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE Transactions on Neural Networks and Learning Systems* **29**, 4065–4076.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J. et al. (2013). Ad click prediction: A view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1222–1230.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)* **135**, 370–384.

Neunhoeffer, M. and Sternberg, S. (2019). How cross-validation can go wrong and what to do about it. *Political Analysis* **27**, 101–106.

Nguyen, H. M., Cooper, E. W. and Kamei, K. (2012). A comparative study on sampling techniques for handling class imbalance in streaming data. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, 1762–1767. IEEE.

Owen, A. B. (2007). Infinitely imbalanced logistic regression. *The Journal of Machine Learning Research* **8**, 761–773.

Rosenblatt, J. D. and Nadler, B. (2016). On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA* **5**, 379–404.

Shao, J. (2003). *Mathematical Statistics*. Springer Science & Business Media.

Shi, P., Wang, P. and Zhang, H. (2019). Distributed logistic regression for separated massive data. In *CCF Conference on Big Data*, 285–296. Springer.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Sun, Y., Kamel, M. S., Wong, A. K. and Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* **40**, 3358–3378.

Sun, Y., Wong, A. K. and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence* **23**, 687–719.

Wang, H. (2019). More efficient estimation for logistic regression with optimal subsamples. *The Journal of Machine Learning Research* **20**, 1–59.

Wang, H. (2020). Logistic regression for massive data with rare events. In *Proceedings of the 37th International Conference on Machine Learning* **PMLR 119**, 9829–9836.

Wang, H., Gao, Y., Shi, Y. and Wang, H. (2016). A fast distributed classification algorithm for large-scale imbalanced data. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1251–1256. IEEE.

Wang, H., Zhang, A. and Wang, C. (2021). Nonuniform negative sampling and log odds correction with rare events data. *Advances in Neural Information Processing Systems* **34**, 19847–19859.

Wang, H., Zhu, R. and Ma, P. (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association* **113**, 829–844.

Yu, J., Wang, H., Ai, M. and Zhang, H. (2022). Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association* **117**, 265–276.

Zhang, H. and Chen, S. X. (2020). Concentration inequalities for statistical inference. *arXiv:2011.02258*.

Zhang, Y., Duchi, J. C. and Wainwright, M. J. (2013). Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research* **14**, 3321–3363.

Zhao, Z.-Q., Zheng, P., Xu, S.-T. and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems* **30**, 3212–3232.

Zhu, M., Su, W. and Chipman, H. A. (2006). Lago: A computationally efficient approach for statistical detection. *Technometrics* **48**, 193–205.

Zhu, X., Li, F. and Wang, H. (2021). Least-square approximation for a distributed system. *Journal of Computational and Graphical Statistics* **30**, 1004–1018.

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B. and Hu, S. (2016). Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2110–2118.

Zhuang, J., Cai, J., Wang, R., Zhang, J. and Zheng, W. (2019). CARE: Class attention to regions of lesion for classification on imbalanced data. In *Proceedings of the 2nd International Conference on Medical Imaging with Deep Learning* **PMLR 102**, 588–597.

Zuo, L., Zhang, H., Wang, H. and Sun, L. (2021). Optimal subsample selection for massive logistic regression with distributed data. *Computational Statistics* **36**, 2535–2562.

Xuetong Li

Department of Business Statistics and Econometrics, Peking University, Beijing 100871, China.

E-mail: 2001110929@stu.pku.edu.cn

Xuening Zhu

School of Data Science, Fudan University, Shanghai 200433, China.

E-mail: xueningzhu@fudan.edu.cn

Hansheng Wang

Department of Business Statistics and Econometrics, Peking University, Beijing 100871, China.

E-mail: hansheng@gsm.pku.edu.cn

(Received July 2022; accepted April 2023)