The Annals of Statistics
2020, Vol. 48, No. 3, 1770–1788
https://doi.org/10.1214/19-AOS1867

LOCAL UNCERTAINTY SAMPLING FOR LARGE-SCALE MULTICLASS LOGISTIC REGRESSION

BY LEI HAN[1], KEAN MING TAN[2], TING YANG[3] AND TONG ZHANG[4]

[1]*Tencent AI Lab, Tencent Technology (Shenzhen) Company Limited, leihan.cs@gmail.com*
[2]*Department of Statistics, University of Michigan, keanming@umich.edu*
[3]*Yelp Inc., ting.yang.pku@gmail.com*
[4]*Department of Mathematics and Department of Computer Science, Hong Kong University of Science and Technology, tongzhang@tongzhang-ml.org*

A major challenge for building statistical models in the big data era is that the available data volume far exceeds the computational capability. A common approach for solving this problem is to employ a subsampled dataset that can be handled by available computational resources. We propose a general subsampling scheme for large-scale multiclass logistic regression and examine the variance of the resulting estimator. We show that asymptotically, the proposed method always achieves a smaller variance than that of the uniform random sampling. Moreover, when the classes are conditionally imbalanced, significant improvement over uniform sampling can be achieved. Empirical performance of the proposed method is evaluated and compared to other methods via both simulated and real-world datasets, and these results match and confirm our theoretical analysis.

1. Introduction. In recent years, there has been an exponential growth in data volume, and this has created demands for building statistical methods to analyze huge datasets. Generally, the size of these datasets far exceeds the available computational capability at hand. For instance, it may not be computationally feasible to perform standard statistical procedures on a single machine when the datasets are huge. Although one remedy is to develop sophisticated distributed computing systems that can directly analyze large datasets, the increased system complexity makes this approach not suitable for all scenarios. Another remedy is to employ a subsampled dataset such that standard statistical methods can be fit by existing computational resources. In fact, such an approach is widely used to solve big data problems. However, subsampling may suffer from loss of statistical accuracy, that is, the variance of the resulting estimator may be large. Therefore, a natural approach is to tradeoff statistical accuracy for computational efficiency by designing an effective sampling scheme that minimizes the reduction of statistical accuracy, given a certain computational capacity.

In this paper, we examine a subsampling approach for solving large-scale multiclass logistic regression problems that are common in practical applications. The general idea of subsampling is to assign an acceptance probability to each data point, and then select observations according to the assigned probabilities. After the subsampling procedure, only a small portion of the data is selected from the full dataset. Hence, the model built using the subsampled data will not be as accurate as that of using the full data. The key challenge is to design a good sampling scheme together with the corresponding estimation procedure such that the loss of statistical accuracy is minimized, given some fixed computational resource. Here, the required computational resource can be measured by the amount of subsampled data.

Received September 2018.
MSC2010 subject classifications. Primary 62D05; secondary 62F10, 62F12.
Key words and phrases. Sampling, large-scale, multiclass logistic regression.

There has been substantial work on subsampling methods for large-scale statistical estimation problems [6–8, 18, 25–27]. The most naive method is to subsample the data uniformly. However, uniform subsampling assigns the same acceptance probability to every data point, which fails to differentiate the importance among the samples. For example, one scenario often encountered in practical applications of logistic regression is when the class labels are imbalanced. This problem has attracted significant interests in the machine learning literature (see survey papers in [5, 12]). Generally, there are two types of commonly encountered class imbalance situations: *marginal imbalance* and *conditional imbalance*. In the case of marginal imbalance, some classes are much rarer than the other classes. This situation often occurs in applications such as the fraud and intrusion detection [1, 14], disease diagnoses [24] and protein fold classification [22]. On the other hand, conditional imbalance describes the case when the labels (denoted as y) for most observations (denoted as $\boldsymbol{x}$) are easy to predict. This happens in applications with very accurate classifiers such as handwriting digits recognition [16] and web/email spam filtering [9, 23]. Note that marginal imbalance implies conditional imbalance, while the reverse is not necessarily true.

For marginally imbalanced binary classification problems, case-control (CC) subsampling, which uniformly selects an equal number of samples from each class, has been widely used in practice in epidemiology and social science studies [17]. Under this scheme, the same amount of samples are subsampled from each class, and thus the sampled data is marginally balanced. It is known that case-control subsampling is more efficient than uniform subsampling when the datasets are marginally imbalanced. However, since the acceptance probability relies on the response variable, the distribution of the subsampled data is skewed by the sample selection process [4]. Correction methods are usually needed to adjust the selection bias [2, 15]. Another method to remove bias in CC subsampling is to weight each sampled data point by the inverse of its acceptance probability. This is known as the weighted case-control method, which has been shown to be consistent and unbiased [13], but it may increase the variance of the resulting estimator [19–21].

One drawback of the standard CC subsampling is that it does not consider the case when the data are conditionally imbalanced. To address this issue, [9] proposed an improved subsampling scheme, referred to as *Local Case-Control* (LCC) sampling, for *binary* logistic regression. The LCC method assigns each data point an acceptance probability determined not only by its label but also by its observed covariates. LCC assigns more importance on data points that are easy to be misclassified according to a consistent pilot estimator, which is an approximate conditional probability estimator possibly obtained using a small amount of uniformly sampled data. The method in [9] fits a logistic model with the sampled data by LCC, and then apply a post-estimation correction to the resulting estimator using the pilot estimator. Therefore, the LCC sampling approach belongs to the correction based methods such as that of [2, 15]. It is shown in [9] that given a consistent pilot estimator, the LCC estimator is consistent with an asymptotic variance that may significantly outperform that of the uniform sampling and CC based sampling methods when the data is strongly conditionally imbalanced.

In this paper, we propose an effective sampling strategy for large-scale *multiclass* logistic regression problems that generalizes the LCC sampling in several aspects. The proposed sampling based estimation procedure follows a framework that can be summarized in the following two steps:

(1) assign an acceptance probability to each data point and select data points according to the assigned probabilities;

(2) fit a multiclass logistic model with the selected data points to obtain an estimate of the unknown model parameter.

In the proposed procedure, the acceptance probability for sampling each data point can be assigned using an *arbitrary* probability function. Unlike correction based methods [9, 15] that are specialized for certain models such as linear model used in LCC sampling, we propose a maximum likelihood estimate (MLE) that integrates the correction into the MLE formulation. This approach allows us to deal with arbitrary sampling probability and always produces a consistent estimator within the original model family, as long as the underlying logistic model is correctly specified. This new integrated estimation method *avoids* the post-estimation correction step used in the existing literature, and hence it can deal with general logistic models such as deep neural networks.

Based on this estimation framework, we propose a new sampling scheme that generalizes LCC sampling. We briefly describe this scheme in the following. Given a rough but consistent prediction $\tilde{p}(y|\boldsymbol{x})$ of the true probability distribution $p(y|\boldsymbol{x})$, this scheme preferentially chooses data points with labels that are conditionally uncertain given their local observations $\boldsymbol{x}$ based on $\tilde{p}(\cdot|\cdot)$. The proposed sampling strategy is thus referred to as *Local Uncertainty Sampling* (LUS). We show that the LUS estimator can achieve an asymptotic variance that is never worse than that of the uniform random sampling. That is, we can achieve variance of no more than γ ($\gamma \geq 1$) times the variance of the full-sample based MLE by using no more than $1/\gamma$ of the sampled data in expectation. Moreover, the required sample size can be significantly smaller than $1/\gamma$ of the full data when the accuracy of the rough estimate $\tilde{p}(y|\boldsymbol{x})$ is high. This generalizes the result for LCC in [9], which reaches a similar conclusion for binary logistic regression when $\gamma \geq 2$.

We also study the case when the model is misspecified. In this case, for binary classification, LUS has the same properties as those of LCC that the subsampling based estimator is consistent to the best estimator for the original population given a consistent pilot estimate (to the best estimator). Unfortunately, for general multiclass problems, the LUS estimator is biased. Nevertheless, we empirically find that the LUS method works well for both binary classification and multiclass classification problems even when the model is misspecified.

We conduct extensive empirical evaluations on both simulated and real-world datasets, showing that the experimental results match the theoretical conclusions and the LUS method significantly outperforms the previous approaches.

Our main contributions can be summarized as follows:

- we propose a general estimation framework for large-scale multiclass logistic regression, which can be used with arbitrary sampling probabilities. The procedure always generates a consistent estimator within the original model family when the model is correctly specified. This method can be applied to general logistic models (e.g., deep neural networks) without the need of post-estimation corrections;
- under this framework, we propose an efficient sampling scheme, referred to as local uncertainty sampling. For any scalar $\gamma \geq 1$, the proposed method can achieve asymptotic variance no more than that of the uniform subsampling with sampling probability $1/\gamma$, using an expected sample size no more than that of the uniform subsampling. Moreover, the required sample size can be significantly smaller than that of the uniform subsampling when the classification accuracy of the underlying problem is high.

2. Preliminaries on multiclass logistic regression. For a K-class classification problem, we observe data point $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \{1, 2, \ldots, K\}$ from an unknown underlying distribution $\mathcal{D}$, where $\boldsymbol{x}$ is the observation vector and y is the corresponding label. Given a set of n independently drawn data points $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$ from $\mathcal{D}$, the goal is to estimate K conditional probabilities $\mathbb{P}_{\mathcal{D}}(Y = k|\boldsymbol{X} = \boldsymbol{x})$ for $k = 1, 2, \ldots, K$. We consider a multi-class

logistic model with the following parametric form:

$$\mathbb{P}_{\mathcal{D}}(Y=k|\boldsymbol{X}=\boldsymbol{x})=\frac{e^{f(\boldsymbol{x},\boldsymbol{\theta}_k)}}{1+\sum_{k'=1}^{K-1}e^{f(\boldsymbol{x},\boldsymbol{\theta}_{k'})}} \quad \text{for } k=1,\ldots,K-1,$$

$$\mathbb{P}_{\mathcal{D}}(Y=K|\boldsymbol{X}=\boldsymbol{x})=\frac{1}{1+\sum_{k'=1}^{K-1}e^{f(\boldsymbol{x},\boldsymbol{\theta}_{k'})}},$$

where $\boldsymbol{\theta}_k$ is the model parameter for the kth class and f is a known function. The above model implies that

$$f(\boldsymbol{x},\boldsymbol{\theta}_k)=\log\frac{\mathbb{P}_{\mathcal{D}}(Y=k|\boldsymbol{X}=\boldsymbol{x})}{\mathbb{P}_{\mathcal{D}}(Y=K|\boldsymbol{X}=\boldsymbol{x})} \quad \text{for } k=1,\ldots,K-1. \tag{2.1}$$

Let $\boldsymbol{\Theta}=(\boldsymbol{\theta}_1^\top,\boldsymbol{\theta}_2^\top,\ldots,\boldsymbol{\theta}_{K-1}^\top)^\top$ be the parameter for the entire model. Equation (2.1) is specified in terms of $K-1$ log-odds with the constraint that the probabilities of all of the classes sum to one. Note that the logistic model uses one reference class as the denominator in the odds-ratios, and the choice of the denominator is arbitrary since the estimates are equivalent under this choice. Throughout the paper, we use the Kth class as the reference class.

When the underlying model is correctly specified, there exists a true parameter $\boldsymbol{\Theta}^0=(\boldsymbol{\theta}_1^{0\top},\boldsymbol{\theta}_2^{0\top},\ldots,\boldsymbol{\theta}_{K-1}^{0\top})^\top$ such that

$$f(\boldsymbol{x},\boldsymbol{\theta}_k^0)=\log\frac{\mathbb{P}_{\mathcal{D}}(Y=k|\boldsymbol{X}=\boldsymbol{x})}{\mathbb{P}_{\mathcal{D}}(Y=K|\boldsymbol{X}=\boldsymbol{x})}. \tag{2.2}$$

Moreover, $\boldsymbol{\Theta}^0$ is the maximizer of the expected population likelihood:

$$L(\boldsymbol{\Theta})=\mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}}\left[\sum_{k=1}^{K-1}\mathbb{I}(y=k)\cdot f(\boldsymbol{x},\boldsymbol{\theta}_k)-\log\left(1+\sum_{k=1}^{K-1}e^{f(\boldsymbol{x},\boldsymbol{\theta}_k)}\right)\right], \tag{2.3}$$

where $\mathbb{I}(\cdot)$ is an indicator function. In the maximum likelihood formulation of multiclass logistic regression, the unknown parameter $\boldsymbol{\Theta}^0$ is estimated from the data by maximizing the empirical likelihood:

$$\hat{L}_n(\boldsymbol{\Theta})=\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{K-1}\mathbb{I}(y_i=k)\cdot f(\boldsymbol{x}_i,\boldsymbol{\theta}_k)-\log\left(1+\sum_{k=1}^{K-1}e^{f(\boldsymbol{x}_i,\boldsymbol{\theta}_k)}\right)\right]. \tag{2.4}$$

For large-scale multiclass logistic regression problems, n can be extremely large. In such cases, solving the multiclass logistic regression problem (2.4) may be computationally infeasible due to the limited computational resources. To overcome this computational challenge, we propose a subsampling framework in the following section.

3. A framework of subsampling based estimation. We propose a novel subsampling based estimation framework for large-scale multiclass logistic regression. Our proposal consists of two main steps:

(1) suppose that an arbitrary sampling probability function $a(\boldsymbol{x},y)\in[0,1]$ is given for every data point $(\boldsymbol{x},y)$. For each pair of observation and label, that is, $(\boldsymbol{x}_i,y_i)$ $(i=1,\ldots,n)$, generate a random binary variable $z_i\in\{0,1\}$, drawn from the $\{0,1\}$-valued Bernoulli distribution $\mathcal{B}(\boldsymbol{x}_i,y_i)$ with acceptance probability

$$\mathbb{P}_{\mathcal{B}(\boldsymbol{x}_i,y_i)}(z_i=1)=a(\boldsymbol{x}_i,y_i);$$

(2) keep samples with $z_i = 1$ for $i \in \{1, \dots, n\}$. Then fit a multiclass logistic regression model based on the selected samples by solving the following optimization problem:

$$\max_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{n} z_i \Bigg[\sum_{k=1}^{K-1} \mathbb{I}(y_i = k) f(\boldsymbol{x}_i, \boldsymbol{\theta}_k) - \log\Bigg(1 + \sum_{k=1}^{K-1} \frac{a(\boldsymbol{x}_i, k)}{a(\boldsymbol{x}_i, K)} e^{f(\boldsymbol{x}_i, \boldsymbol{\theta}_k)}\Bigg)\Bigg]. \tag{3.1}$$

We now derive the above procedure under the assumption that the logistic model is correctly specified as in Equation (2.2). As we will show later, the acceptance probability used in the first step can be an arbitrary function, and the above method always produces a consistent estimator for the original population. The computational complexity in the second step is reduced to fitting the model with only $\sum_{i=1}^{n} z_i$ samples after the subsampling step.

Given $(\boldsymbol{x}, y) \sim \mathcal{D}$, we draw a random variable z according to the Bernoulli distribution $\mathcal{B}(\boldsymbol{x}, y)$. This gives the following augmented distribution $\mathcal{A}$ for the joint variable $(\boldsymbol{x}, y, z) \in \mathbb{R}^d \times \{1, 2, \dots, K\} \times \{0, 1\}$ with probability

$$\begin{aligned}&\mathbb{P}_{\mathcal{A}}(\boldsymbol{X} = \boldsymbol{x}, Y = k, Z = z)\\ &\quad = \mathbb{P}_{\mathcal{D}}(\boldsymbol{X} = \boldsymbol{x}, Y = k)\big[a(\boldsymbol{x}, k)\mathbb{I}(z = 1) + \big(1 - a(\boldsymbol{x}, k)\big)\mathbb{I}(z = 0)\big].\end{aligned}$$

Note that each sampled data point follows $(\boldsymbol{x}_i, y_i) \sim \mathcal{D}$, and the random variable z_i is independently drawn from $\mathcal{B}(\boldsymbol{x}_i, y_i)$. It follows that each joint data point $(\boldsymbol{x}_i, y_i, z_i)$ is drawn i.i.d. from the distribution $\mathcal{A}$. For the sampled data point $(\boldsymbol{x}_i, y_i)$ with $z_i = 1$, the conditional distribution of the random variable $(\boldsymbol{x}, y)$ is

$$\mathbb{P}_{\mathcal{A}}(\boldsymbol{X} = \boldsymbol{x}, Y = k | Z = 1) \propto \mathbb{P}_{\mathcal{D}}(\boldsymbol{X} = \boldsymbol{x}, Y = k) a(\boldsymbol{x}, k).$$

Therefore, we have

$$\log \frac{\mathbb{P}_{\mathcal{A}}(Y = k | \boldsymbol{X} = \boldsymbol{x}, Z = 1)}{\mathbb{P}_{\mathcal{A}}(Y = K | \boldsymbol{X} = \boldsymbol{x}, Z = 1)} = \log \frac{\mathbb{P}_{\mathcal{D}}(Y = k | \boldsymbol{X} = \boldsymbol{x})}{\mathbb{P}_{\mathcal{D}}(Y = K | \boldsymbol{X} = \boldsymbol{x})} + \log \frac{a(\boldsymbol{x}, k)}{a(\boldsymbol{x}, K)}.$$

If f is correctly specified for $\mathcal{D}$, then the following function family

$$g(\boldsymbol{x}, \boldsymbol{\theta}_k) = f(\boldsymbol{x}, \boldsymbol{\theta}_k) + \log \frac{a(\boldsymbol{x}, k)}{a(\boldsymbol{x}, K)} \tag{3.2}$$

is correctly specified for $\mathcal{A}$, that is, the true parameter $\boldsymbol{\Theta}^0$ in equation (2.2) also satisfies

$$g(\boldsymbol{x}, \boldsymbol{\theta}_k^0) = \log \frac{\mathbb{P}_{\mathcal{A}}(Y = k | \boldsymbol{X} = \boldsymbol{x}, Z = 1)}{\mathbb{P}_{\mathcal{A}}(Y = K | \boldsymbol{X} = \boldsymbol{x}, Z = 1)} \quad \text{for } k = 1, \dots, K - 1.$$

Therefore, we have the following logistic model under $\mathcal{A}$:

$$\mathbb{P}_{\mathcal{A}}(Y = k | \boldsymbol{X} = \boldsymbol{x}, Z = 1) = \frac{e^{g(\boldsymbol{x}, \boldsymbol{\theta}_k)}}{1 + \sum_{k=1}^{K-1} e^{g(\boldsymbol{x}, \boldsymbol{\theta}_k)}} \quad \text{for } k = 1, \dots, K - 1,$$

$$\mathbb{P}_{\mathcal{A}}(Y = K | \boldsymbol{X} = \boldsymbol{x}, Z = 1) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{g(\boldsymbol{x}, \boldsymbol{\theta}_k)}}.$$

Then, given an arbitrary sampling probability function $a(\boldsymbol{x}, y) \in [0, 1]$, $\boldsymbol{\Theta}^0$ can be obtained by using MLE with respect to the new population $\mathcal{A}$:

$$\max_{\boldsymbol{\Theta}} R(\boldsymbol{\Theta}) := \mathbb{E}_{\boldsymbol{x}, y, z \sim \mathcal{A}}\, z \Bigg[\sum_{k=1}^{K-1} \mathbb{I}(y = k) \cdot g(\boldsymbol{x}, \boldsymbol{\theta}_k) - \log\Bigg(1 + \sum_{k=1}^{K-1} e^{g(\boldsymbol{x}, \boldsymbol{\theta}_k)}\Bigg)\Bigg]. \tag{3.3}$$

In practice, the model parameter $\boldsymbol{\Theta}^0$ can be estimated by empirical conditional MLE with respect to the sampled data $\{(\boldsymbol{x}_i, y_i, z_i) : i = 1, \ldots, n\}$:

$$\max_{\boldsymbol{\Theta}} \hat{R}_n(\boldsymbol{\Theta}) := \frac{1}{n}\sum_{i=1}^{n} z_i \left[\sum_{k=1}^{K-1} \mathbb{I}(y_i = k) \cdot g(\boldsymbol{x}_i, \boldsymbol{\theta}_k) - \log\left(1 + \sum_{k=1}^{K-1} e^{g(\boldsymbol{x}_i, \boldsymbol{\theta}_k)}\right)\right], \tag{3.4}$$

which is equivalent to problem (3.1). Let $\hat{\boldsymbol{\Theta}}_{\text{Sub}} = \arg\max_{\boldsymbol{\Theta}} \hat{R}_n(\boldsymbol{\Theta})$ be the subsampling based estimator. As we will see in the next section, $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ is a consistent estimator of $\boldsymbol{\Theta}^0$ when the model is correctly specified.

4. Asymptotic analysis. In this section, we examine the asymptotic behavior of the proposed method described in Section 3. All of the proofs are provided in Appendix A in the supplementary material [11].

4.1. *Asymptotic properties.* First, based on the empirical likelihood in equation (3.4), we have the following result for $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$.

THEOREM 4.1 (Consistency and asymptotic normality). *Suppose that the parameter space is compact that* $\mathbb{P}_{\mathcal{D}}(f(\boldsymbol{x}, \boldsymbol{\Theta}) \neq f(\boldsymbol{x}, \boldsymbol{\Theta}^0)) > 0$ *for all* $\boldsymbol{\Theta} \neq \boldsymbol{\Theta}^0$ *and* $\boldsymbol{x} \sim \mathcal{D}$. *Moreover, assume the quantities* $\|\nabla_{\boldsymbol{\theta}_k} f(\boldsymbol{x}, \boldsymbol{\theta}_k)\|$, $\|\nabla^2_{\boldsymbol{\theta}_k} f(\boldsymbol{x}, \boldsymbol{\theta}_k)\|$ *and* $\|\nabla^3_{\boldsymbol{\theta}_k} f(\boldsymbol{x}, \boldsymbol{\theta}_k)\|$ *for* $k = 1, \ldots, K-1$ *are bounded under some norm* $\|\cdot\|$ *for any* $\boldsymbol{\Theta}$.[1] *Let* $p(\boldsymbol{x}, k) = \mathbb{P}_{\mathcal{D}}(Y = k | \boldsymbol{X} = \boldsymbol{x})$. *If equation (2.2) is satisfied, that is, the model is correctly specified, then given an arbitrary sampling probability function* $a(\boldsymbol{x}, y)$, *as* $n \to \infty$, *the following holds*:

(1) $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ *converges to* $\boldsymbol{\Theta}^0$;
(2) $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ *asymptotically follows the normal distribution*:

$$\sqrt{n}(\hat{\boldsymbol{\Theta}}_{\text{Sub}} - \boldsymbol{\Theta}^0) \xrightarrow{d} \mathcal{N}(\mathbf{0}, [\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \nabla \mathbf{S} \nabla^\top]^{-1}), \tag{4.1}$$

where $\nabla = \text{diag}([\nabla_{\boldsymbol{\theta}_1} f(\boldsymbol{x}, \boldsymbol{\theta}_1^0), \nabla_{\boldsymbol{\theta}_2} f(\boldsymbol{x}, \boldsymbol{\theta}_2^0), \ldots, \nabla_{\boldsymbol{\theta}_{K-1}} f(\boldsymbol{x}, \boldsymbol{\theta}_{K-1}^0)])$ *is a block diagonal matrix, each block of which is* $\nabla_{\boldsymbol{\theta}_k} f(\boldsymbol{x}, \boldsymbol{\theta}_k^0)$, *and*

$$\mathbf{S} = \text{diag}\left(\begin{bmatrix} a_1 p_1 \\ a_2 p_2 \\ \vdots \\ a_{K-1} p_{K-1} \end{bmatrix}\right) - \frac{1}{\sum_{k=1}^{K} a_k p_k} \begin{bmatrix} a_1 p_1 \\ a_2 p_2 \\ \vdots \\ a_{K-1} p_{K-1} \end{bmatrix} \begin{bmatrix} a_1 p_1 \\ a_2 p_2 \\ \vdots \\ a_{K-1} p_{K-1} \end{bmatrix}^\top, \tag{4.2}$$

with notation a_k *and* p_k *indicating* $a(\boldsymbol{x}, k)$ *and* $p(\boldsymbol{x}, k)$, *respectively.*

Theorem 4.1 shows that given an arbitrary sampling probability $a(\boldsymbol{x}, k)$, the proposed method in Section 3 generates a consistent estimator $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ *without* post-estimation correction. This is different from existing methods such as the LCC sampling method in [9], which employs post-estimation corrections. One benefit of the proposed framework is that without post-estimation correction, we can still produce a consistent estimator in the original model family, and our framework allows different sampling functions for different data points

[1] This assumption can be relaxed with some more complex arguments, for example, by using the moment assumptions on the derivatives.

$(\boldsymbol{x}_i, y_i)$. For example, in time series analysis, we may want to sample recent data points more aggressively than past data points. This can be naturally handled by the above framework, but not by the post-estimation correction based approach. Another benefit is that the framework can be naturally applied to models with regularization, which can be regarded as a restriction on the parameter space of $\boldsymbol{\Theta}$.

From Theorem 4.1, the estimator $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ is asymptotically normal with variance $[\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\nabla \times \mathbf{S}\nabla^\top]^{-1}$. Although the sampling probability $a(\boldsymbol{x}, y)$ can be arbitrary, it is important to select an effective sampling probability such that the variance of the resulting estimator is as small as possible. In the following, we study a specific choice of $a(\boldsymbol{x}, y)$ that achieves this purpose.

4.2. *Sampling strategy.* Recall from the definition of $\mathbf{S}$ equation (4.2). Let $\mathbf{S}_{\text{full}}$ be the corresponding matrix $\mathbf{S}$ when we set $a(\boldsymbol{x}, k) = 1$ for all k, that is, we accept all data points in the dataset. Then

$$\mathbf{S}_{\text{full}} = \operatorname{diag}\left(\begin{bmatrix} p(\boldsymbol{x},1) \\ p(\boldsymbol{x},2) \\ \vdots \\ p(\boldsymbol{x},K-1) \end{bmatrix}\right) - \begin{bmatrix} p(\boldsymbol{x},1) \\ p(\boldsymbol{x},2) \\ \vdots \\ p(\boldsymbol{x},K-1) \end{bmatrix}\begin{bmatrix} p(\boldsymbol{x},1) \\ p(\boldsymbol{x},2) \\ \vdots \\ p(\boldsymbol{x},K-1) \end{bmatrix}^\top. \tag{4.3}$$

If we set $a(\boldsymbol{x}, k) = \frac{1}{\gamma}$ for all k for some $\gamma \geq 1$, that is, we sample a fraction of $\frac{1}{\gamma}$ of the full dataset uniformly at random, we denote the corresponding matrix as $\mathbf{S}_{\text{US}:\frac{1}{\gamma}}$. Then $\mathbf{S}_{\text{US}:\frac{1}{\gamma}} = \frac{1}{\gamma}\mathbf{S}_{\text{full}}$. In the following, we denote the asymptotic variances of $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$, the full-sample based estimator and the estimator obtained from $\frac{1}{\gamma}$ uniformly sampled data as

$$\mathcal{V}_{\text{Sub}} = \left[\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\nabla\mathbf{S}\nabla^\top\right]^{-1},\ \mathcal{V}_{\text{full}} = \left[\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\nabla\mathbf{S}_{\text{full}}\nabla^\top\right]^{-1}$$

and

$$\mathcal{V}_{\text{US}:\frac{1}{\gamma}} = \gamma\mathcal{V}_{\text{full}},$$

respectively. Our purpose is to find a better sampling strategy with lower variance than that of uniform sampling. That is, we want to choose an acceptance probability function $a(\boldsymbol{x}, k)$ such that

$$\mathcal{V}_{\text{Sub}} \preceq \gamma\mathcal{V}_{\text{full}} = \mathcal{V}_{\text{US}:\frac{1}{\gamma}}$$

for some $\gamma \geq 1$, under the constraint that

$$\mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}}a(\boldsymbol{x}, y) \leq \frac{1}{\gamma}.$$

The constraint requires the expected subsample size to be no more than n/γ, that is, we sample no more than $1/\gamma$ fraction of the full data.

THEOREM 4.2 (Sampling strategy). *For any observation* $\boldsymbol{x}$, *let*

$$q(\boldsymbol{x}) = \max\bigl(0.5, p(\boldsymbol{x}, 1), \ldots, p(\boldsymbol{x}, K)\bigr). \tag{4.4}$$

Given any $\gamma \geq 1$, *consider the following choice of the acceptance probability function*:

(1) *for* $\gamma \geq 2q(\boldsymbol{x})$, *set* $a(\boldsymbol{x}, k)$ *as*

$$a(\boldsymbol{x}, k) = \begin{cases} \dfrac{2(1-q(\boldsymbol{x}))}{\gamma} & \text{if } p(\boldsymbol{x}, k) = q(\boldsymbol{x}) \geq 0.5, \\ \dfrac{2q(\boldsymbol{x})}{\gamma} & \text{otherwise}, \end{cases} \qquad k = 1, \ldots, K; \tag{4.5}$$

(2) *for* $1 \le \gamma < 2q(\boldsymbol{x})$, *set* $a(\boldsymbol{x}, k)$ *as*

$$a(\boldsymbol{x}, k) = \begin{cases} \dfrac{1 - q(\boldsymbol{x})}{\gamma - q(\boldsymbol{x})} & \text{if } p(\boldsymbol{x}, k) = q(\boldsymbol{x}) \ge 0.5, \\ 1 & \text{otherwise}, \end{cases} \quad k = 1, \ldots, K. \tag{4.6}$$

Under the same assumptions and definitions in Theorem 4.1, *we have*

$$\mathcal{V}_{\text{Sub}} \preceq \gamma \mathcal{V}_{\text{full}} = \mathcal{V}_{\text{US}:\frac{1}{\gamma}}, \tag{4.7}$$

and the expected number of subsampled examples is

$$n_{\text{Sub}} = n\mathbb{E}_{\boldsymbol{x}, y \sim \mathcal{D}} a(\boldsymbol{x}, y) \le \frac{n}{\gamma}. \tag{4.8}$$

It is easy to check that the assigned acceptance probability in Theorem 4.2 is always valid, that is, it is a value in [0, 1]. With the sampling strategy in Theorem 4.2, we always use no more than a fraction of $1/\gamma$ of the full data to obtain an estimator with variance no more than γ times the variance of the full-sample based MLE. In other words, the proposed method is never worse than the uniform sampling method. Moreover, the required sample size n_{Sub} can be significantly smaller than n/γ under favorable conditions.

More precisely, we have the following formula for the expected conditional sampling probability:

$$\mathbb{E}_{\boldsymbol{x}, y \sim \mathcal{D}}\, a(\boldsymbol{x}, y) = \begin{cases} \dfrac{4}{\gamma} q(\boldsymbol{x})\big(1 - q(\boldsymbol{x})\big) & \text{under case (1) in Theorem 4.2,} \\ \dfrac{\gamma(1 - q(\boldsymbol{x}))}{\gamma - q(\boldsymbol{x})} & \text{under case (2) in Theorem 4.2.} \end{cases}$$

Therefore, in favorable cases in which most $q(\boldsymbol{x}) \approx 1$ for $\boldsymbol{x} \sim \mathcal{D}$, that is, the data are conditionally imbalanced, our method will subsample very few samples to achieve the desired variance compared to that of the uniform random sampling with a rate of $1/\gamma$.

An intuitive explanation for the proposed sampling strategy is that if there exists a class k that dominates the other classes for any given $\boldsymbol{x}$, that is, $p(\boldsymbol{x}, k) \ge 0.5$, then the sampling probability will be proportional to "$1 - accuracy$". That is, when the classification accuracy for the underlying problem is high, the proposed sampling method will significantly outperform the uniform sampling.

For binary classification problem ($K = 2$), Theorem 4.2 reduces to the LCC sampling in [9] when $\gamma \ge 2$. Although a method to achieve a desired variance for the case of $\gamma \in [1, 2)$ was also discussed in [9], it is different from our sampling strategy. Moreover, there is no evidence for that method that the number of samples needed by LCC for achieving such a desired variance is never worse than that of the uniform sampling for the case of $\gamma \in [1, 2)$. In fact, the empirical performance of LCC can be worse than our method under the case of $\gamma \in [1, 2)$, as we will show in the experimental section.

In the multiclass case, our method is not a natural extension of the LCC sampling, which would imply a method to set all class probabilities to $1/K$ after sampling. Instead, we will assign a smaller sampling probability for $(\boldsymbol{x}, y)$ when $p(\boldsymbol{x}, y) \ge 0.5$. The method is less likely to select a sample when the label y coincides with the prediction of the underlying true model, while a sample will likely be selected if y contradicts the underlying true model. Since the sampling strategy prefers data points with uncertain labels, we refer to it as *Local Uncertainty Sampling* (LUS). In the following, we will indicate the estimator as $\hat{\boldsymbol{\Theta}}_{\text{LUS}}$ if the acceptance probability is set according to equations (4.5) and (4.6) (recall that we use the notation $\hat{\boldsymbol{\Theta}}_{\text{Sub}}$ with an arbitrary sampling function $a(\boldsymbol{x}, y)$).

4.3. *Randomness on the acceptance probability.* In Theorem 4.2, $a(\boldsymbol{x}, k)$ is a function of the true probability $p(\boldsymbol{x}, k)$ for $k = 1, \ldots, K$. In practice, $p(\boldsymbol{x}, k)$ is unknown. We instead use a roughly estimated probability $\tilde{p}(\boldsymbol{x}, k)$ to compute the acceptance probability, which we refer to as the *pilot* estimate.

There are multiple ways to estimate $\tilde{p}(\boldsymbol{x}, k)$. For example, one can obtain a pilot estimator $\boldsymbol{\lambda}$ by fitting the model using a smaller independent data set, or obtain $\tilde{p}(\boldsymbol{x}, k)$ from a different and simpler parametric model or non-parametric methods. This is different from the LCC method, which involves post-estimation correction that relies on an explicit pilot estimator $\boldsymbol{\lambda}$ that is additive to the original model parameter. For simplicity, in the following analysis, we assume that $\tilde{p}(\boldsymbol{x}, k)$ is computed from a pilot estimator $\boldsymbol{\lambda}$.

Recall from Theorem 4.1 that given an acceptance probability, the asymptotic variance for $\hat{\boldsymbol{\Theta}}_{\mathrm{LUS}}$ is dependent on $\mathbf{S}$ and $\boldsymbol{\nabla}$. Since $a(\boldsymbol{x}, k)$ is computed based on $\boldsymbol{\lambda}$, from equation (4.2), we see that $\mathbf{S}$ is now a function of both $\boldsymbol{\lambda}$ and the model parameter $\boldsymbol{\Theta}$. We rewrite $\mathbf{S}$ as $\mathbf{S}(\boldsymbol{\lambda}, \boldsymbol{\Theta})$ to emphasize its dependency on $\boldsymbol{\lambda}$ and $\boldsymbol{\Theta}$. Similarly, we represent $\boldsymbol{\nabla}$ as $\boldsymbol{\nabla}(\boldsymbol{\Theta})$. Moreover, with the use of the pilot estimator, from equation (3.3), $R(\cdot)$ is now a function of $\boldsymbol{\lambda}$ and $\boldsymbol{\Theta}$, and we rewrite it as $R(\boldsymbol{\lambda}, \boldsymbol{\Theta})$.

In the following, we characterize the asymptotic distribution of $\hat{\boldsymbol{\Theta}}_{\mathrm{LUS}}$ when a pilot estimator $\hat{\boldsymbol{\lambda}}$ is estimated from an independent data set.

COROLLARY 4.1. *Let $\tilde{p}(\boldsymbol{x}, k)$ be a probability estimate computed using a pilot estimator $\hat{\boldsymbol{\lambda}}$. If $\hat{\boldsymbol{\lambda}} \xrightarrow{p} \boldsymbol{\Theta}^0$ such that $\tilde{p}(\boldsymbol{x}, k) \xrightarrow{p} p(\boldsymbol{x}, k)$, we have*

$$\sqrt{n}(\hat{\boldsymbol{\Theta}}_{\mathrm{LUS}} - \boldsymbol{\Theta}^0) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathcal{V}_{\mathrm{LUS}}),$$

where $\mathcal{V}_{\mathrm{LUS}} = [\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boldsymbol{\nabla}(\boldsymbol{\Theta}^0)\mathbf{S}(\boldsymbol{\Theta}^0, \boldsymbol{\Theta}^0)\boldsymbol{\nabla}(\boldsymbol{\Theta}^0)^\top]^{-1}$ is a constant that is independent of the pilot estimator $\hat{\boldsymbol{\lambda}}$.

Corollary 4.1 implies that as long as the pilot estimator $\hat{\boldsymbol{\lambda}}$ is a consistent estimator of $\boldsymbol{\Theta}^0$ such that $\tilde{p}(\boldsymbol{x}, k)$ converges to $p(\boldsymbol{x}, k)$, the randomness induced by the pilot estimator $\hat{\boldsymbol{\lambda}}$ will not inflate the asymptotic variance of $\hat{\boldsymbol{\Theta}}_{\mathrm{LUS}}$. The result is in conjunction with that of LCC when the model is correctly specified, and is also a generalization of their results to the case of $K > 2$.

4.4. *Model misspecification.* In practice, the model in equation (2.2) may not be correctly specified, so a true parameter $\boldsymbol{\Theta}^0$ may not exist. Under such case, we denote the best estimator obtained by maximizing equation (2.3) for the original population $\mathcal{D}$ as $\boldsymbol{\Theta}^*$ and denote the corresponding probability estimate as $p^*(\boldsymbol{x}, k)$ for $k = 1, \ldots, K$. Since the model is misspecified, we know that $p^*(\boldsymbol{x}, k)$ may not equal to $p(\boldsymbol{x}, k)$.

In the following, we study the properties of LUS and consider the cases when $K = 2$ and $K > 2$ separately. To distinguish between the two cases, we use $\boldsymbol{\theta}$ to denote the model parameter for $K = 2$, that is, binary classification problems. The following proposition summarizes results for $K = 2$.

PROPOSITION 4.1. *For $K = 2$, suppose that the parameter space of $\boldsymbol{\theta}$ is compact that $\mathbb{P}_{\mathcal{D}}(f(\boldsymbol{x}, \boldsymbol{\theta}) \neq f(\boldsymbol{x}, \boldsymbol{\theta}^*)) > 0$ for all $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$ and $\boldsymbol{x} \sim \mathcal{D}$. Under model misspecification, let $\boldsymbol{\lambda} = \boldsymbol{\theta}^*$ and we have*

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} R(\boldsymbol{\theta}^*, \boldsymbol{\theta}).$$

Moreover, $\boldsymbol{\theta}^*$ is the unique maximizer of $R(\boldsymbol{\theta}^*, \boldsymbol{\theta})$ and the LUS estimator is a consistent estimator of $\boldsymbol{\theta}^*$.

Proposition 4.1 implies that under model misspecification, if we use a perfect pilot estimate $\boldsymbol{\lambda} = \boldsymbol{\theta}^*$ to compute the acceptance probability, then $\hat{\boldsymbol{\theta}}_{\mathrm{LUS}}$ would also converge to $\boldsymbol{\theta}^*$. Again, in practice, we never know $\boldsymbol{\theta}^*$ a priori. Therefore, we will need a pilot estimator $\hat{\boldsymbol{\lambda}}$ estimated from an independent data set. In the following proposition, we study how the randomness of $\hat{\boldsymbol{\lambda}}$ affects the performance of the LUS estimator when the model is misspecified.

PROPOSITION 4.2. *For $K = 2$, assume that the pilot estimator $\hat{\boldsymbol{\lambda}}$ follows $\sqrt{n}(\hat{\boldsymbol{\lambda}} - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathcal{V}_\lambda)$ for some $\mathcal{V}_\lambda \succ 0$. Set the acceptance probability according to equations (4.5) and (4.6). Under the same conditions in Proposition 4.1, and under model misspecification, we have*

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{LUS}} - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}\big(\mathbf{0}, \mathbb{H}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)^{-1}\big[J(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) + C(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)\mathcal{V}_\lambda C(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)^\top\big] \times \mathbb{H}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)^{-1}\big),$$

where

$$\mathbb{H}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = -\nabla^2_{\boldsymbol{\theta}} R(\boldsymbol{\lambda}, \boldsymbol{\theta}),\ J(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \mathrm{Var}\big(\sqrt{n}\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\lambda}, \boldsymbol{\theta})\big),\ C(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{\partial^2 R(\boldsymbol{\lambda}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\lambda}}.$$

Proposition 4.2 implies that as long as the pilot estimator is consistent to $\boldsymbol{\theta}^*$, $\hat{\boldsymbol{\theta}}_{\mathrm{LUS}}$ is consistent to $\boldsymbol{\theta}^*$. Moreover, under model misspecification, we see that there is an inflation in the asymptotic variance induced by the random pilot estimator $\hat{\boldsymbol{\lambda}}$. From Propositions 4.1 and 4.2, when the model is misspecified for $K = 2$, the LUS method has the same properties as those of the LCC method proposed in [9].

However, when $K > 2$, the LUS estimator is biased.

PROPOSITION 4.3. *Under model misspecification, when $K > 2$, the LUS estimator is biased even when $\boldsymbol{\lambda} = \boldsymbol{\Theta}^*$. However, if there exists a class k such that $p(\boldsymbol{x}, \boldsymbol{\theta}_k^*) \geq 0.5$, then $\boldsymbol{\theta}_k^* = \arg\max_{\boldsymbol{\theta}_k} R(\boldsymbol{\Theta}^*, \boldsymbol{\Theta}^*_{-k})$ where $\boldsymbol{\Theta}^*_{-k}$ is the parameter by setting $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^*$ for $i \neq k$.*

Proposition 4.3 implies that when $\boldsymbol{\lambda} = \boldsymbol{\Theta}^*$, if there exists a majority class such that $p(\boldsymbol{x}, \boldsymbol{\theta}_k^*) \geq 0.5$, then the LUS estimator with respect to the kth class will be consistent to $\boldsymbol{\theta}_k^*$ when the other parameters $\boldsymbol{\theta}_i$ are fixed as $\boldsymbol{\theta}_i^*$ for $i \neq k$. Explicit characterizations of the bias for the LUS estimator when $K > 2$ are complex and we put them in Appendix B in the supplementary material [11].

Nevertheless, in our empirical studies, we find that the LUS method works well for both binary classification and multiclass classification problems, even under model misspecification.

5. Local uncertainty sampling algorithm. In order to apply the sampling strategy in Theorem 4.2 empirically, according to Corollary 4.1, the main idea is to employ a rough but consistent estimate $\tilde{p}(\boldsymbol{x}, k)$ of the probabilities $p(\boldsymbol{x}, k)$ given $\boldsymbol{x}$, and then assign the acceptance probability according to $\tilde{p}(\boldsymbol{x}, k)$. In fact, we only need to have an approximate estimate of $q(\boldsymbol{x})$ according to equation (4.4); that is, we only need to roughly estimate the probability of the majority class given $\boldsymbol{x}$. In practice, $\tilde{p}(\boldsymbol{x}, k)$ can be obtained via a consistent pilot estimator $\boldsymbol{\lambda}$ where $\tilde{p}(\boldsymbol{x}, k) = e^{f(\boldsymbol{x}, \boldsymbol{\lambda}_k)}/(1 + \sum_{k'}^{K-1} e^{f(\boldsymbol{x}, \boldsymbol{\lambda}_{k'})})$. This is similar to what is employed in [9]. However, it is not a necessity in our case to compute a pilot estimator $\boldsymbol{\lambda}$ in the exact original model family, because we do not need post-correction with a consistent specific model form for both the pilot estimator and the sampling based estimator. As shown in our

Algorithm 1 The LUS Algorithm for Multiclass Logistic Regression

1: Choose a desired $\gamma \geq 1$ to achieve the desired variance.
2: Given a rough estimate $\tilde{\boldsymbol{p}}_i = (\tilde{p}_{i,1}, \ldots, \tilde{p}_{i,K})^\top$ for each data point $\boldsymbol{x}_i$, where $\tilde{p}_{i,k}$ is a roughly estimated probability of $\boldsymbol{x}_i$ belonging to class k.
3: Scan the data once and generate the random variables $z_i \sim \text{Bernoulli}(a(\boldsymbol{x}_i, y_i) : z_i = 1)$ based on the acceptance probability $a(\boldsymbol{x}_i, y_i)$ defined as

$$a(\boldsymbol{x}_i, y_i) = \begin{cases} \dfrac{1 - \tilde{q}_i}{\gamma - \max(\tilde{q}_i, 0.5\gamma)} & \text{if } \tilde{p}_{i,y_i} = \tilde{q}_i \geq 0.5, \\ \min(1, 2\tilde{q}_i/\gamma) & \text{otherwise,} \end{cases}$$

where $\tilde{q}_i = \max(0.5, \tilde{p}_{i,1}, \ldots, \tilde{p}_{i,K})$.
4: Fit a multiclass logistic regression model to the subsample set $\{(\boldsymbol{x}_i, y_i) : z_i = 1\}$ with the model function g defined in equation (3.2):

$$\hat{\boldsymbol{\Theta}}_{LUS} = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{n} z_i \left(\sum_{k=1}^{K-1} \mathbb{I}(y_i = k) \cdot g(\boldsymbol{x}_i, \boldsymbol{\theta}_k) - \log\left(1 + \sum_{k=1}^{K-1} e^{g(\boldsymbol{x}_i, \boldsymbol{\theta}_k)}\right)\right). \tag{5.1}$$

5: Output $\hat{\boldsymbol{\Theta}}_{\text{LUS}}$.

MNIST experiment below, we use a simpler neural network structure to obtain the pilot estimate compared to the network used for the logistic regression. Moreover, one may use other techniques such as neighborhood based methods [3] to obtain a pilot estimator.

In many applications, a rough estimate is often easy to obtain. For example, when data arrives in a data streaming manner, a pilot estimator trained using previous observations can be used to fit a new model when new observations arrive. Moreover, a rough estimate obtained on a small subset of the full population can be used for training on the entire dataset. In our experiments, we adopt the latter: a small uniformly subsampled subset of the original population is used to obtain the rough estimate $\tilde{p}(\boldsymbol{x}, k)$. This choice is sufficient for our method to obtain good practical performance in our numerical studies. Moreover, in addition to using a small subset of the data for the pilot, we can adopt a simpler model as described in the last paragraph at the same time, as we will show in the MNIST experiment below. The LUS algorithm is summarized in Algorithm 1.

6. Experiments. In this section, we evaluate the performance of the LUS method and compare it to the uniform sampling (US) and case-control (CC) sampling methods on both simulated and real-world datasets. For the CC sampling method, we extend the standard CC sampling considered in the binary classification problem to multiclass problem by sampling equal number of data points for each class. Under marginal imbalance, to sample a desired total amount of data points, if some minority classes do not have enough samples, we keep all data for those classes and subsample equal number of the remaining data points from other classes. In addition, we also compare the LUS and LCC methods on the Web Spam dataset, which is a binary classification problem studied in [9]. The experiments are implemented on a single machine with 2.2 GHz quad-core Intel Core i7 and 16 GB memory.

6.1. *Simulation*: *Marginal imbalance*. We first simulate the case when the data is marginally imbalanced. We generate a 3-class Gaussian model according to $(\boldsymbol{X}|Y = k) \sim$

$\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, which we denote as $\mathcal{D}$ to indicate the true data generating distribution. We set the number of features as $d = 20$,

$$\boldsymbol{\mu}_1 = [\underbrace{1, 1, \ldots, 1}_{10}, \underbrace{0, 0, \ldots, 0}_{10}]^\top, \ \boldsymbol{\mu}_2 = [\underbrace{0, 0, \ldots, 0}_{10}, \underbrace{1, 1, \ldots, 1}_{10}]^\top,$$

and

$$\boldsymbol{\mu}_3 = [\underbrace{0, 0, \ldots, 0}_{20}]^\top.$$

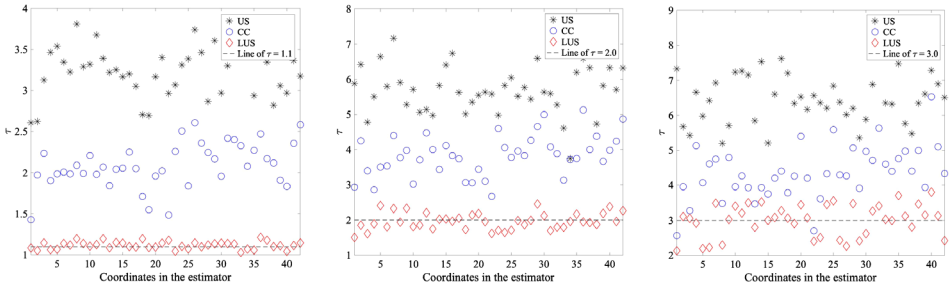
The covariance matrices for classes $k = 1, 2, 3$ are assigned to be the same, that is, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \mathbf{I}_d$, where $\mathbf{I}_d$ is a $d \times d$ identity matrix. So the true log-odds function f is linear and we use a linear model to fit the simulated data, that is, the model is correctly specified. Moreover, we set $\mathbb{P}(Y = 1) = 0.1$, $\mathbb{P}(Y = 2) = 0.8$ and $\mathbb{P}(Y = 3) = 0.1$. This implies that the data is marginally imbalanced and the second class dominates the population.

Since the true data distribution $\mathcal{D}$ is known in this case, we directly generate the full dataset from the distribution $\mathcal{D}$. For the full dataset, we generate $n = 50{,}000$ data points. The entire procedure is repeated for 200 times to obtain the variance of different estimators. For the LUS method, we randomly generate $n_{\text{pilot}} = 5000$ data points (i.e., an amount of 10% of the full data) from $\mathcal{D}$ to obtain a pilot estimator in *every* repetition. Moreover, we generate another $n_{\text{test}} = 100{,}000$ data points to test the prediction accuracy of different methods.

Recall that γ controls the desired variance of the LUS estimator according to Theorem 4.2. In the following experiments, we will test different values of $\gamma = \{1.1, 1.2, \ldots, 1.9, 2, 3, 4, 5\}$, respectively. Given the value of γ, suppose the LUS method will subsample a number of n_{Sub} data points. Then we let the US and CC sampling methods select an amount of $n_{\text{Sub}} + n_{\text{pilot}}$ examples to achieve fair comparison, because the LUS method has to pay for its usage of a random pilot estimate.

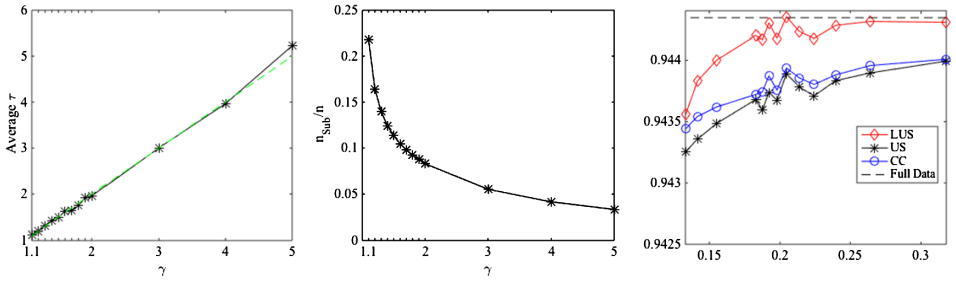
Since $\boldsymbol{\theta}_k \in \mathbb{R}^d$ $(k = 1, \ldots, K - 1)$ in this case and there is an additional intercept parameter, the estimator contains a total number of $(d + 1)(K - 1)$ coordinates. Denote τ as the coordinatewise ratio between the variance of the coordinate in the subsampling based estimator and the variance of the coordinate in the full-sample based MLE. We show the τ value for each coordinate under different values of $\gamma = \{1.1, 2, 3\}$ in Figure 1. In this simulation, there are 42 coordinates. From the figures, we observe that the τ value for each coordinate of the LUS method is around γ, which matches our theoretical analysis in Theorem 4.2. On the other hand, the variances of the US and CC sampling methods are much higher than that of the LUS method, even if US and CC sample n_{pilot} more data points than the LUS method.

In Figure 2(a), we plot the relationship between the average τ for all coordinates against γ. From the figure, we observe that the relationship is close to the $y = x$ line (the dashed green



(a) $\gamma = 1.1$ (b) $\gamma = 2$ (c) $\gamma = 3$

FIG. 1. *The τ value for each coordinate under different values of γ. τ denotes the ratio between the variance of each coordinate in the subsampling based estimator and the variance of the coordinate in the full-sample based MLE, that is, $\tau = \text{Var}(\hat{\theta}_{\text{Sub}})/\text{Var}(\hat{\theta}_{\text{full}})$.*



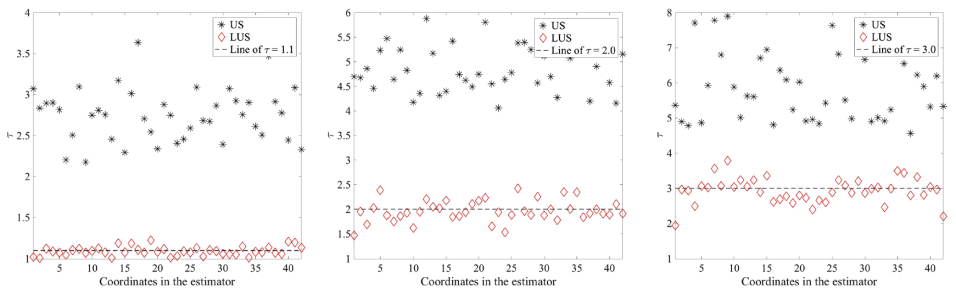
(a) Average τ v.s. γ (b) n_{Sub}/n v.s. γ (c) Accuracy v.s. $\frac{n_{Sub}+n_{pilot}}{n}$

FIG. 2. *Plots in the first simulation.*

line), which implies that τ approximately equals γ. These experimental results well match our theoretical analyses. Figure 2(b) reports the relationship between n_{Sub}/n and γ. Figure 2(c) shows the relationship between the prediction accuracy on the test data and the proportion of used training data $(n_{\text{Sub}} + n_{\text{pilot}})/n$. From the figure, when $(n_{\text{Sub}} + n_{\text{pilot}})/n$ decreases, the prediction accuracy of all the methods decreases, while the LUS method shows much slower degradation compared to the US and CC methods. Moreover, according to Figure 2(c), we only need about 20% of the full data (including those used for computing the pilot estimator) to achieve the same prediction accuracy as the full-sample based MLE, implying that the LUS method is very effective for reducing the computational cost while preserving high accuracy.

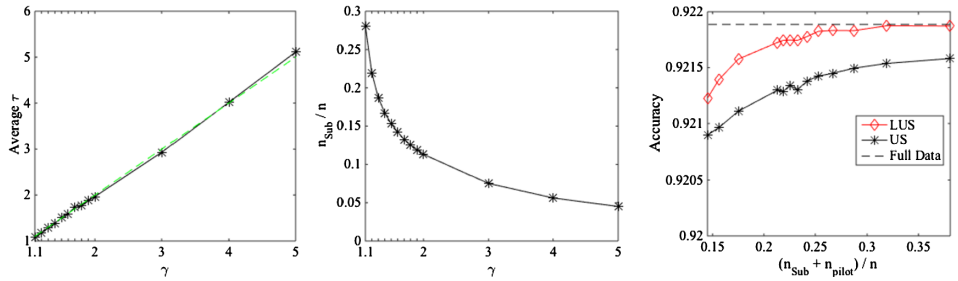
6.2. *Simulation*: *Marginal balance*. In this section, we generate marginally balanced data with conditional imbalance. Under this situation, the CC sampling method is identical to US, and hence we omit the CC sampling method from our comparison. The settings are exactly the same as those in the previous simulation, except that we let $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 2) = \mathbb{P}(Y = 3) = \frac{1}{3}$, which implies that the data is marginally balanced. However, as we will see later, this simulated data is conditionally imbalanced.

The τ value for each coordinate when $\gamma = \{1.1, 2, 3\}$ is shown in Figure 3. The relationship between the average τ for all the coordinates and γ is plotted in Figure 4(a). Figure 4(b) reports the relationship between $(n_{\text{Sub}} + n_{\text{pilot}})/n$ and γ. In Figure 4(c), we show the relationship between the prediction accuracy on the test data and the proportion of used training data $(n_{\text{Sub}} + n_{\text{pilot}})/n$. The results are similar to those of the previous simulation and demonstrate the effectiveness of the LUS method under the marginally balanced (but conditionally imbalanced) case. Figure 4(c) suggests that we only need about 25% of the full data (including those used for computing the pilot estimator) to achieve the same prediction accuracy as that of the full-sample based MLE.



(a) $\gamma = 1.1$ (b) $\gamma = 2$ (c) $\gamma = 3$

FIG. 3. *The τ value for each coordinate under different values of γ.* $\tau = \text{Var}(\hat{\theta}_{\text{Sub}})/\text{Var}(\hat{\theta}_{\text{full}})$.



(a) Average τ v.s. γ (b) n_{Sub}/n v.s. γ (c) Accuracy v.s. $\frac{n_{Sub}+n_{pilot}}{n}$

FIG. 4. *Plots in the second simulation.*

In the simulations, we have fixed the amount of data used for computing the pilot estimate as $\frac{n_{\text{pilot}}}{n} = 10\%$. In general, increasing or decreasing this amount will reduce or increase the variance of the pilot estimate and affect the performance of the LUS estimator accordingly. We provide simulations to show how the performance of the LUS estimator changes with respect to n_{pilot} in Appendix C in the supplementary material [11], and the results suggest that a small n_{pilot} is sufficient for LUS to achieve good performance.

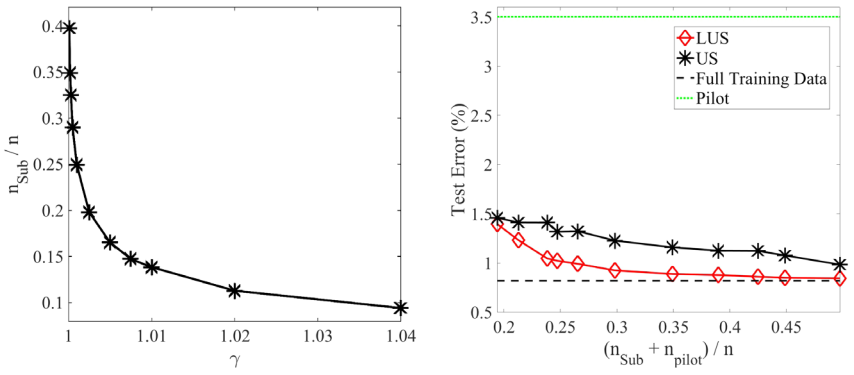
In large-scale applications, an empirical trick with respect to the pilot estimate is to use two or more phases for computing the pilot estimate. That is, we can recursively apply an LUS estimator as the pilot estimator, and recompute the LUS estimator. Since the variance of the LUS estimator is smaller than that of the uniform sampling based estimator (when they subsample the same amount of data), using multiple phases will reduce the variance of the final estimator compared with using uniform sampling as the pilot estimator.

Despite of these flexibilities on choosing the pilot estimate, for simplicity, we will still use uniform sampling and fix the portion of data for the pilot estimate in the following experiments for numerical evaluations.

6.3. *MNIST data.* In this section, we evaluate different methods on the MNIST data,[2] which is a benchmark dataset in image classification problems and the state-of-the-art results have achieved less than 1% test error rate on this dataset. Therefore, the classification accuracy of this problem is high. Note that different from the LCC sampling which uses linear models, our LUS method can accommodate general logistic models. In this experiment, we let the model function f of the LUS estimator to be one of the state-of-the-art deep neural networks. Since we have no knowledge about the underlying true model in this real dataset, the adopted neural network might be misspecified. In order to save computational cost, we use a simpler neural net structure with fewer parameters to obtain the pilot estimate $\tilde{p}(\boldsymbol{x}, k)$. It is worth mentioning that this simpler neural network is different from the one used for the final LUS estimator, because the LUS method only requires a rough estimate $\tilde{p}(\boldsymbol{x}, k)$ instead of an explicit estimator which is needed by post-estimation correction based methods. The detailed network structures and parameter settings are provided in Appendix D in the supplementary material [11]. For the US method, we apply the same network structure used by the final LUS estimator to achieve fair comparison. Since the MNIST data is marginally balanced, the CC method is equivalent to the US method and we omit its comparison here.

The training set consists of 60,000 images and the test set has 10,000 images. We uniformly select $n_{\text{pilot}} = 6000$ data points (i.e., 10% of the training data) to compute the rough estimate $\tilde{p}(\boldsymbol{x}, k)$ in *every* repetition and perform 10 repetitions of the experiment to obtain the average performance of different methods. Similar to the simulations, we assume the LUS method

[2] http://yann.lecun.com/exdb/mnist/



(a) n_{Sub}/n v.s. γ (b) Test error v.s. $\frac{n_{Sub}+n_{pilot}}{n}$

FIG. 5. *Plots in MNIST data.*

samples a number of n_{Sub} data points and let the US method sample $n_{\text{Sub}} + n_{\text{pilot}}$ data points. Note that the setting is a bit unfair for LUS because the n_{pilot} data points used for computing $\tilde{p}(\boldsymbol{x}, k)$ are processed by a simpler neural network. However, we still keep this protocol in the experiment.

We test a number of values of γ in the range $(1, 1.04]$ and Figure 5(a) plots the proportion of used data $(n_{\text{Sub}} + n_{\text{pilot}})/n$ against γ. Figure 5(c) shows the relationship between the test error (%) and $(n_{\text{Sub}} + n_{\text{pilot}})/n$. Note that the rough pilot estimate has a relatively large error rate of about 3.5%; this is due to the fact that it employs a simpler network structure to save computational cost. Nevertheless, the LUS method can achieve an error rate below 1% using only about 25% of the training data (10% for the pilot estimate and 15% for LUS); with about 45% of the training data (10% for the pilot estimate and 35% for LUS), it achieves the same error rate as that obtained by using the full training data and the standard neural network. The LUS method consistently outperforms the US method. Table 1 shows the speedup of the LUS method compared to the full-sample based estimation.

6.4. *Web Spam data*: *Binary classification*. In this section, we compare the LUS method with the LCC method on the Web Spam data,[3] which is a binary classification problem used in [9] to evaluate the LCC method. Since the comparison among the LCC, US and CC methods on this dataset has been reported in [9], we do not repeat them here and focus on the comparison between the LUS and LCC methods. The Web Spam data contains 350,000 web pages and about 60% of them are web spams. This dataset is approximately marginally balanced, but it has been shown to have strong conditional imbalance in [9]. We adopt the same settings as described in [9] to compare the LUS and the LCC methods. That is, we use linear logistic model and select 99 features which appear in at least 200 documents, and the features are log-transformed. 10% of the observations are uniformly selected to obtain a pilot

TABLE 1

Speedup of the LUS method on MNIST data when using 45% *of the training set* (10% *for the pilot estimate and* 35% *for LUS*), *that is*, *achieving the same error rate with the full-training-sample based MLE*

	The pilot estimate	LUS	Full training data	Speedup
Seconds	51.0	369.2	1115.1	2.7

[3] http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html

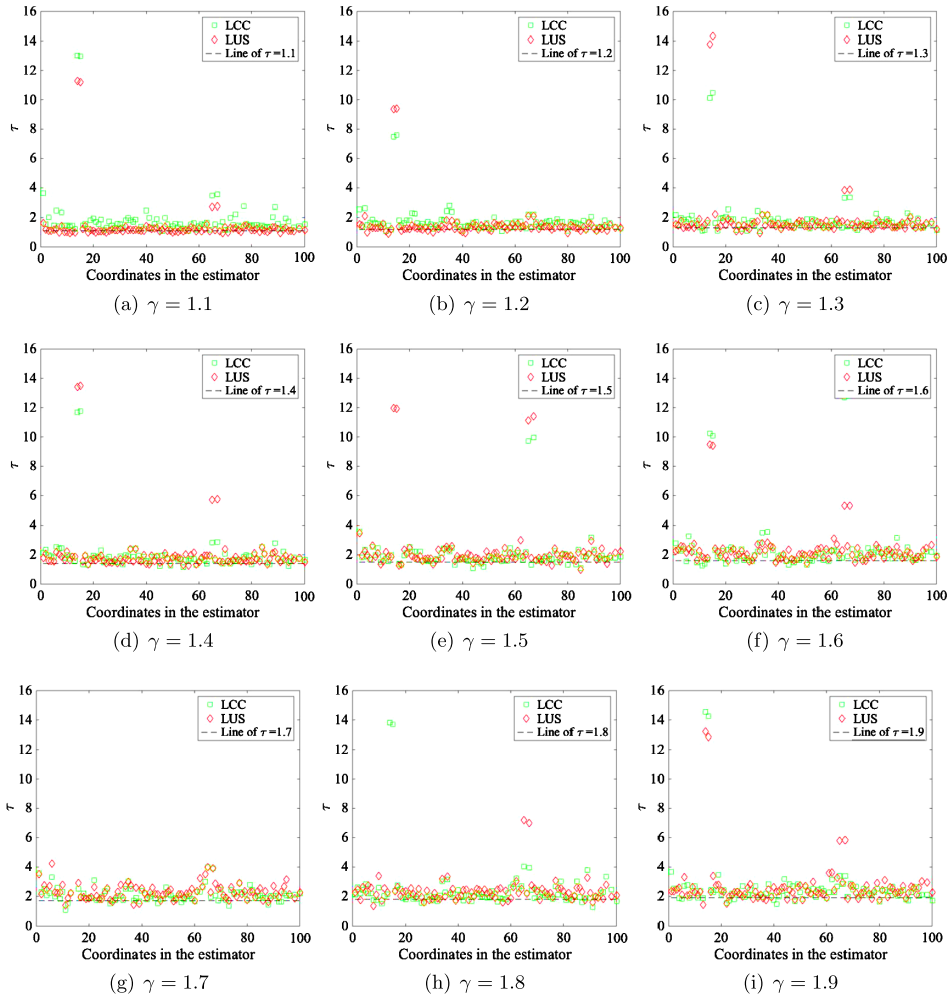


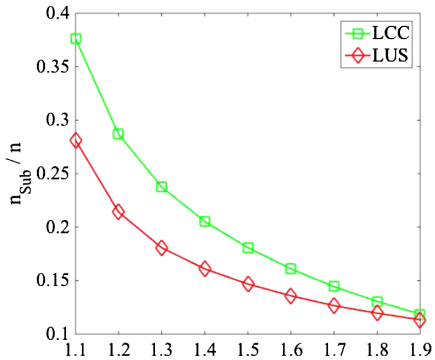
FIG. 6. *The τ value for each coordinate with respect to different values of γ in the Web Spam data, where* $\tau = \mathrm{Var}(\hat{\theta}_{\mathrm{sub}})/\mathrm{Var}(\hat{\theta}_{\mathrm{full}})$.

estimator. Since we only have a single dataset, we follow [9] to uniformly subsample 100 datasets, each of which contains 100,000 data points, as 100 independent "full" datasets, and then repeat the experiments 100 times for comparison.

Observe that when $\gamma \geq 2$, the LUS and LCC methods are equivalent to each other by setting the parameter $c = \frac{1}{\gamma - 1} \leq 1$ of LCC in Section 3.3 of [9]. Therefore, we only focus on the case of $1 \leq \gamma < 2$. In the first experiment, similar to previous experiments, we test different values of $\gamma = \{1.1, 1.2, \ldots, 1.9\}$ and accordingly set $c = \{10, 5, \ldots, \frac{10}{9}\}$ for LCC, so that the two methods have the same asymptotic variance. Then we will compare the number of subsampled data points to see which method is more effective in terms of subsampled data size n_{Sub}.

Figure 6 plots the τ values for different choices of γ. As expected from the theoretical results, both the LUS and LCC methods have the same variance that is approximately γ (or $1 + \frac{1}{c}$) times variance of the full-sample based MLE. Next, we compare the subsampling proportion of different methods when γ changes in Figure 7. From the figure, the LUS method consistently subsamples fewer data points compared with LCC when they achieve the same variance as shown in Figure 6.

Alternatively, we fix the proportion of the sampled examples n_{Sub}/n for both LUS and LCC methods (by carefully choosing γ and c), and we test the variance of the estimators



(a) n_{Sub}/n v.s. γ

FIG. 7. n_{Sub}/n *v.s.* γ *(or* $1+\frac{1}{c}$*) in Web Spam data.*

to see which one has lower variance. Table 2 shows the average variance of the coordinates in the LCC estimator and the LUS estimator. We observe that the LUS estimator always achieves lower variance compared to that of the LCC estimator. These results demonstrate that the LUS method is not only theoretically better justified but also more effective than the LCC method in practice for the case of $\gamma \in [1, 2)$.

7. Conclusion. This paper introduced a general subsampling method for solving large-scale multiclass logistic regression problems. We investigated the asymptotic variance of the proposed estimator when the model is correctly specified. Based on the theoretical analysis, we proposed an effective sampling strategy, referred to as local uncertainty sampling, to achieve any given level of desired variance. We proved that the method always achieves lower variance than random subsampling for a given expected sample size, and the improvement may be significant under the favorable condition of strong conditional imbalance. Therefore, the method can effectively accelerate the computation of large-scale multiclass logistic regression in practice. Empirically, we will need a pilot estimate of the probability to set up the acceptance probability. We proved that the variance of the proposed estimator is independent of the randomness of this pilot estimate as long as the pilot estimator is consistent.

We also studied the case of model misspecification. We showed that for binary classification problems ($K = 2$), the proposed method can generate a consistent estimator (to the best estimator of the misspecified model) if the pilot estimator is consistent. For $K > 2$, the proposed estimator is biased and we also provided analysis to quantify the bias.

The empirical studies supported the theory and demonstrated that the local uncertainty sampling method outperforms the uniform sampling, case-control sampling and the local case-control sampling methods under various settings. By using the proposed method, we are able to select a very small subset of the original data to achieve the same performance as that

TABLE 2
Average variance of the coordinates in the estimators of LCC and LUS

	n_{Sub}/n			
	0.15	0.2	0.25	0.3
LCC	2.2511 ± 1.9586	1.8691 ± 1.3655	1.8270 ± 1.4078	1.7303 ± 1.1001
LUS	2.1108 ± 2.0495	1.5608 ± 1.1668	1.6017 ± 1.5674	1.6196 ± 1.7184

of the full dataset, which provides an effective mean for big data computation under limited resources.

This work suggests several future directions. First, as we have mentioned at the end of Section 6.2, one can iteratively apply an obtained LUS estimator as the pilot estimator for the next round of fitting the model. This is closely related to the boosting method [10] and a deep discussion on this relationship would be of great interest. Second, considering a situation of online learning with limited budget, the LUS method would likely provide an effective sampling strategy for this problem. Moreover, in high-dimensional settings, where sparse models, for example, Lasso and Group Lasso, are widely adopted, it would be interesting to extend the LUS method to deal with regularized multiclass logistic regression with special considerations on high-dimensional asymptotic regime.

Acknowledgments. We would like to thank the reviewers for their valuable comments.

This work was supported in part by NSF Grants IIS-1250985 and IIS-1407939, NSF Grant DMS-1811315 and NIH Grant R01AI116744.

SUPPLEMENTARY MATERIAL

Supplement: Proofs and supplementary experimental results (DOI: 10.1214/19-AOS1867SUPP; .pdf). Appendix A in the supplementary material contains proofs for Theorem 4.1, Theorem 4.2, Corollary 4.1, Proposition 4.1, Proposition 4.2 and Proposition 4.3. Appendix B provides explicit characterizations of the bias for the LUS estimator when $K > 2$ under model misspecification. Appendix C provides some additional simulation results for varying the amount of the data used for the pilot estimate. Finally, Appendix D reports the detailed neural network structures used in the MNIST dataset.

REFERENCES

[1] ABE, N., ZADROZNY, B. and LANGFORD, J. (2004). An iterative method for multi-class cost-sensitive learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 3–11.

[2] ANDERSON, J. A. (1972). Separate sample logistic discrimination. *Biometrika* **59** 19–35. MR0345332 https://doi.org/10.1093/biomet/59.1.19

[3] ATKESON, C. G., MOORE, A. W. and SCHAAL, S. (1997). Locally weighted learning for control. In *Lazy Learning* 75–113. Springer, Berlin.

[4] BRESLOW, N. (1982). Design and analysis of case-control studies. *Annu. Rev. Public Health* **3** 29–54. https://doi.org/10.1146/annurev.pu.03.050182.000333

[5] CHAWLA, N. V., JAPKOWICZ, N. and KOTCZ, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explor. Newsl.* **6** 1–6.

[6] CORTES, C., MANSOUR, Y. and MOHRI, M. (2010). Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems* 442–450.

[7] CORTES, C., MOHRI, M., RILEY, M. and ROSTAMIZADEH, A. (2008). Sample selection bias correction theory. In *Algorithmic Learning Theory. Lecture Notes in Computer Science* **5254** 38–53. Springer, Berlin. MR2540648 https://doi.org/10.1007/978-3-540-87987-9_8

[8] DHILLON, P., LU, Y., FOSTER, D. P. and UNGAR, L. (2013). New subsampling algorithms for fast least squares regression. In *Advances in Neural Information Processing Systems* 360–368.

[9] FITHIAN, W. and HASTIE, T. (2014). Local case-control sampling: Efficient subsampling in imbalanced data sets. *Ann. Statist.* **42** 1693–1724. MR3257627 https://doi.org/10.1214/14-AOS1220

[10] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting. *Ann. Statist.* **28** 337–407. MR1790002 https://doi.org/10.1214/aos/1016218223

[11] HAN, L., TAN, K. M., YANG, T. and ZHANG, T. (2020). Supplement to "Local uncertainty sampling for large-scale multiclass logistic regression." https://doi.org/10.1214/19-AOS1867SUPP.

[12] HE, H. and GARCIA, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21** 1263–1284.

[13] HORVITZ, D. G. and THOMPSON, D. J. (1952). A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.* **47** 663–685. MR0053460

[14] KIM, H.-C., PANG, S., JE, H.-M., KIM, D. and BANG, S. Y. (2002). Pattern classification using support vector machine ensemble. In *Proceedings of the International Conference on Pattern Recognition* **2** 160–163.

[15] KING, G. and ZENG, L. (2001). Logistic regression in rare events data. *Polit. Anal.* **9** 137–163.

[16] LECUN, Y., BOTTOU, L., BENGIO, Y. and HAFFNER, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* **86** 2278–2324.

[17] MANTEL, N. and HAENSZEL, W. (1959). Statistical aspects of the analysis of data from retrospective studies. *J. Natl. Cancer Inst.* **22** 719–748.

[18] MINEIRO, P. and KARAMPATZIAKIS, N. (2013). Loss-proportional subsampling for subsequent ERM. Preprint. Available at arXiv:1306.1840.

[19] SCOTT, A. and WILD, C. (2002). On the robustness of weighted methods for fitting models to case-control data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **64** 207–219. MR1904701 https://doi.org/10.1111/1467-9868.00333

[20] SCOTT, A. J. and WILD, C. J. (1986). Fitting logistic models under case-control or choice based sampling. *J. Roy. Statist. Soc. Ser. B* **48** 170–182. MR0867995

[21] SCOTT, A. J. and WILD, C. J. (1991). Fitting logistic regression models in stratified case-control studies. *Biometrics* **47** 497–510. MR1132540 https://doi.org/10.2307/2532141

[22] TAN, A. C., GILBERT, D. and DEVILLE, Y. (2003). Multi-class protein fold classification using a new ensemble machine learning approach. *Genome Inform.* **14** 206–217.

[23] WEBB, S., CAVERLEE, J. and PU, C. (2006). Introducing the Webb Spam Corpus: Using email spam to identify Web spam automatically. In *Proceedings of the Third Conference on Email and Anti-Spam*.

[24] WIDODO, A. and YANG, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process.* **21** 2560–2574.

[25] XIE, Y. and MANSKI, C. F. (1989). The logit model and response-based samples. *Sociol. Methods Res.* **17** 283–302.

[26] ZADROZNY, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the International Conference on Machine Learning* 114.

[27] ZHANG, T. and OLES, F. (2000). The value of unlabeled data for classification problems. In *Proceedings of the International Conference on Machine Learning* 1191–1198.