A Review on Optimal Subsampling Methods for Massive Datasets

Yaqiong Yao[1] and HaiYing Wang*[1]

[1] *Department of Statistics, University of Connecticut, Storrs, Connecticut, USA*

Abstract

Subsampling is an effective way to deal with big data problems and many subsampling approaches have been proposed for different models, such as leverage sampling for linear regression models and local case control sampling for logistic regression models. In this article, we focus on optimal subsampling methods, which draw samples according to optimal subsampling probabilities formulated by minimizing some function of the asymptotic distribution. The optimal subsampling methods have been investigated to include logistic regression models, softmax regression models, generalized linear models, quantile regression models, and quasi-likelihood estimation. Real data examples are provided to show how optimal subsampling methods are applied.

Keywords *Asymptotic mean squared error, big data, optimal subsampling*

1 Introduction

As we step into the big data era, more and more attention is focused on how to deal with data with enormous size and complex frame under limited computational resources. In the field of statistics, various techniques were developed to analyze massive datasets, such as divide-and-conquer method (Lin and Xie, 2011), online updating for streaming data (Schifano et al., 2016), stochastic gradient descent (Toulis et al., 2017), random projection (Drineas et al., 2011; Mahoney, 2011) and subsampling (Drineas et al., 2006; Ma et al., 2015; Wang et al., 2018, 2019).

Subsampling method draws a subdata set from the full dataset and estimates the interested parameters by the chosen subdata. The fundamental concern of the subsampling method is how to select the subdata. The more informative observations we choose, the

*Corresponding author. Email: haiying.wang@uconn.edu

better approximation performance we could expect. Hence, uniform subsampling is not preferred because every observations are treated equally no matter how much information one observation carries. For linear regression, leverage sampling has been detailed discussed by Drineas et al. (2006); Mahoney (2011), and was named as algorithm leveraging in Ma et al. (2015); Ma and Sun (2015). The subsamples obtained by this method are drawn from the full dataset with replacement based on the normalized leverage scores or their variants. The asymptotic normality and asymptotic unbiasedness of the leveraging sampling estimator were studied in Ma et al. (2020). The leveraged volume sampling was proposed by Derezinski et al. (2018) for linear regression, which yields an unbiased coefficient estimator and has the same tail bonds as leverage sampling. Besides these probabilistic methods for linear models, a deterministic method named information-based optimal subdata selection (IBOSS) was proposed by Wang et al. (2019) aiming at finding a subdata that has maximal information matrix under D-optimality. This method is also applicable under divide-and-conquer setting, which was discussed in (Wang, 2019a). The IBOSS approach was extended to include the logistic regression in Cheng et al. (2020). The local case control sampling for logistic regression was proposed by Fithian and Hastie (2014), which draws samples by Poisson subsampling and determine whether one observation is in or not in the sample using information from both the response and covariates. By extending the idea of the local case control sampling, a local uncertainty sampling algorithm was introduced by Han et al. (2020) for softmax regression. Pronzato and Wang (2020) proposed an algorithm for steaming dataset where the subdata is selected sequentially based on the estimated quantile.

Optimal subsampling method is a probabilistic approach where subsamples are expected to be drawn based on the optimal subsampling probabilities, which are derived by minimizing the asymptotic covariance matrix of the random sampling based estimators under certain optimality criterion. The optimal subsampling method for logistic regression was introduced by Wang et al. (2018), which formulates the optimal subsampling probabilities by minimizing the asymptotic mean squared error (MSE) of the subsample estimator. Since the expressions of the optimal subsampling probabilities involves the maximum likelihood estimator (MLE) of the full data, the authors proposed a two-stage adaptive algorithm which uses a pilot sample estimator to substitute the full data MLE. This method was named as optimal subsampling methods motivated from the A-optimality criterion (OSMAC), and was improved in Wang (2019b) by adopting unweighted target functions for subsamples and Poisson subsampling. In addition to logistic regression, OSMAC was investigated to include softmax regression (Yao and Wang, 2018), generalized linear models (Ai et al., 2019), quantile regression (Wang and Ma, 2020) and quasi-likelihood (Yu et al., 2020). This article aims at introducing the optimal subsampling method and illustrates its practical implements with the following real data examples in **R** (R Core Team, 2020).

- Income dataset (Dua and Graff, 2017). This dataset was extracted from 1994 Census database and aimed at predicting whether one person's annual income is over 50000 or not based on various personal information such as age, education level, gender and financial situation.
- Bike sharing dataset (Fanaee-T and Gama, 2013). Bike sharing system monitors bike rental situation hourly. It records the hourly weather information and working day information. This dataset intends to modeling the hourly bike rental numbers under different conditions.
- Physicochemical properties of protein tertiary structure dataset (Dua and Graff, 2017). This dataset was extracted from Critical Assessment of protein Structure Prediction and provides information of the protein structure. We are going to model the size of the residue based on the given information, which is ranging from 0 to 22.

The rest of the paper is organized as follows. Section 2 talks about the adaptive optimal subsampling method for logistic regression and softmax regression. Section 3 presents more efficient algorithms for logistic regression by introducing unweighted estimator and Poisson subsampling into the adaptive optimal subsampling method. Section 4 discusses the adaptive optimal subsampling method for generalized linear models. Section 5 shows the application of optimal subsampling for quantile regression. A brief summary is presented in Section 6.

2 Optimal subsampling methods motivated from the A-optimality criterion

Suppose that $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ are N independent and identically distributed observations, where $\boldsymbol{x}_i \in \mathcal{R}^d, i = 1, 2, ..., N$, are covariates, and $y_i, i = 1, 2, ..., N$, are responses. For a logistic regression, $y_i \in \{0, 1\}$ is a binary variable. Given $\boldsymbol{x}_i$, the response y_i satisfies that

$$P(y_i = 1|\boldsymbol{x}_i) = p(\boldsymbol{x}_i, \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})}, \quad i = 1, 2, ..., N,$$

where $\boldsymbol{\beta} \in \mathcal{R}^d$ is the unknown regression coefficient, and can be estimated by the MLE $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$, which is the maximizer of

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left[y_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} - \log\{1 + \exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})\}\right].$$

This optimization problem can be solved by the Newton-Raphson method in $O(\eta N d^2)$ time where η is the number of iterations for the Newton-Raphson method to converge. To reduce the computational burden when N is large, an optimal subsampling method named OSMAC targeting at approximating the full data MLE $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ was proposed in Wang et al. (2018). To begin with, we introduce the general subsample estimator obtained by a subsample drawing from the full dataset with arbitrary subsampling probabilities $\{\pi_i\}_{i=1}^N$ in Algorithm 1.

Algorithm 1 General Subsampling Algorithm

Subsampling with replacement:

- Assign subsampling probabilities $\{\pi_i\}_{i=1}^N$ to each observation.
- Draw n data points with replacement based on $\{\pi_i\}_{i=1}^N$, and denoted the subsample as $\{\boldsymbol{x}_i^*, y_i^*, \pi_i^*\}_{i=1}^n$.

Estimation: Obtain the regression coefficient estimator $\hat{\boldsymbol{\beta}}_{\mathrm{sub}}$ by maximizing

$$\ell^*(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{y_i^*\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^* - \log\{1 + \exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^*)\}}{\pi_i^*}. \tag{1}$$

It has been proved that $\hat{\boldsymbol{\beta}}_{\mathrm{sub}}$ is consistent to $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ and the approximation error $\hat{\boldsymbol{\beta}}_{\mathrm{sub}} - \hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ is asymptotically normal conditional on the full data. The underlying idea of the

OSMAC is to find the optimal subsampling probabilities which minimize the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\beta}}_{\text{sub}} - \hat{\boldsymbol{\beta}}_{\text{MLE}}$, denoted as $\boldsymbol{V}_N$. To compare matrices, A-optimality criterion is adopted, which minimizes the trace of this asymptotic variance-covariance matrix. The optimal subsampling probabilities under A-optimality criterion are

$$\pi_i^{\text{optA}} = \frac{|y_i - p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})| \|\boldsymbol{M}_L^{-1}\boldsymbol{x}_i\|}{\sum_{j=1}^N |y_j - p(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}_{\text{MLE}})| \|\boldsymbol{M}_L^{-1}\boldsymbol{x}_j\|}, \quad i = 1, ..., N, \tag{2}$$

where $\boldsymbol{M}_L = N^{-1}\sum_{i=1}^N p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})\{1 - p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})\}\boldsymbol{x}_i\boldsymbol{x}_i^{\text{T}}$. To reduce the computational burden, L-optimality is also considered, intending to minimize the trace of the asymptotic variance-covariance matrix of $\boldsymbol{M}_L(\hat{\boldsymbol{\beta}}_{\text{sub}} - \hat{\boldsymbol{\beta}}_{\text{MLE}})$. Thus, the L-optimal subsampling probabilities minimize $\text{tr}(\boldsymbol{M}_L^{\text{T}}\boldsymbol{V}_N\boldsymbol{M}_L^{\text{T}})$ and have expressions

$$\pi_i^{\text{optL}} = \frac{|y_i - p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})| \|\boldsymbol{x}_i\|}{\sum_{j=1}^N |y_j - p(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}_{\text{MLE}})| \|\boldsymbol{x}_j\|}, \quad i = 1, ..., N. \tag{3}$$

Both A- and L- optimal subsampling probabilities depend on the responses and covariates, and contain $\hat{\boldsymbol{\beta}}_{\text{MLE}}$, which is the quantity that we are approximating. To solve this problem, a pilot sample estimator is used to substitute $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ in (2) and (3). The pilot sample can be drawn from the full dataset by uniform subsampling or case control subsampling whose subsampling probabilities are $\pi_i^0 = N^{-1}$ and $\pi_i^0 = (2\sum_{i=1}^N y_i)^{-y_i}(2N - 2\sum_{i=1}^N y_i)^{y_i - 1}$, respectively. Furthermore, $\boldsymbol{M}_L$ can be approximated by the pilot sample and pilot sample estimator to reduce the computational complexity. It takes $O(Nd^2)$ time to compute π_i^{optA}, and $O(Nd)$ time to compute π_i^{optL}. The OSMAC is summarized in Algorithm 2.

Theorem 6 in Wang et al. (2018) has proved the asymptotical normality of $\tilde{\boldsymbol{\beta}}^{\text{OS}}$ conditionally on the full data and pilot sample estimator. The convergence rate is at the order of $n_1^{-1/2}$, which is not related to the full data size. This means that even the full data size increases, the information contained by the subsample may not change. In addition, Algorithm 2 is an adaptive algorithm in that the approximately optimal subsample probabilities rely on the pilot sample estimator. Thus an inaccurate pilot sample estimator may affect the accuracy of the final estimator. Algorithm 2 greatly reduces the computational cost compared with the full data computation, but still needs to pro-

Algorithm 2 Adaptive optimal subsampling algorithm for logistic regression

Pilot sampling:

– Run Algorithm 1 with subsample size n_0 and subsampling probabilities π_i^0. Obtain the pilot subsample estimator $\hat{\boldsymbol{\beta}}^{\text{sub},0}$.

– Store the pilot subsample and the corresponding subsampling probabilities $\{\boldsymbol{x}_i^{*0}, y^{*0}, \pi_i^{*0}\}_{i=1}^{n_0}$.

Second step sampling:

– Calculate the approximate optimal subsampling probabilities

$$\hat{\pi}_i^{\text{optA}} = \frac{|y_i - p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{\text{sub},0})|\|\hat{\boldsymbol{M}}_L(\hat{\boldsymbol{\beta}}^{\text{sub},0})^{-1}\boldsymbol{x}_i\|}{\sum_{j=1}^N |y_j - p(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}_{\text{MLE}})|\|\hat{\boldsymbol{M}}_L(\hat{\boldsymbol{\beta}}^{\text{sub},0})^{-1}\boldsymbol{x}_j\|}, \quad \text{or} \tag{4}$$

$$\hat{\pi}_i^{\text{optL}} = \frac{|y_i - p(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{\text{sub},0})|\|\boldsymbol{x}_i\|}{\sum_{j=1}^N |y_j - p(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}_{\text{MLE}})|\|\boldsymbol{x}_j\|} \tag{5}$$

under selected optimality criterion, where

$$\hat{\boldsymbol{M}}_L(\hat{\boldsymbol{\beta}}^{\text{sub},0}) = \frac{1}{n_0 N}\sum_{i=1}^{n_0} \frac{p(\boldsymbol{x}_i^{*0}, \hat{\boldsymbol{\beta}}^{\text{sub},0})\{1 - p(\boldsymbol{x}_i^{*0}, \hat{\boldsymbol{\beta}}^{\text{sub},0})\}\boldsymbol{x}_i^{*0}(\boldsymbol{x}_i^{*0})^{\text{T}}}{\pi_i^{*0}}.$$

– Run Algorithm 1 with subsample size n_1 and subsampling probabilities $\{\hat{\pi}_i^{\text{optA}}\}_{i=1}^N$ or $\{\hat{\pi}_i^{\text{optL}}\}_{i=1}^N$.

– Record the second step subsample and the corresponding subsampling probabilities $\{\boldsymbol{x}_i^{*1}, y_i^{*1}, \pi_i^{*1}\}_{i=1}^{n_1}$.

Estimation: Combine pilot sample and second step sample, and denote the combined sample as $\{\boldsymbol{x}_i^*, y_i^*, \pi_i^*\}_{i=1}^{n_0+n_1}$. Obtain the final estimator $\tilde{\boldsymbol{\beta}}^{\text{OS}}$ by maximizing

$$\ell^*_{\text{sub}}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0+n_1} \frac{y^*\boldsymbol{\beta}^{\text{T}}\boldsymbol{x}_i^* - \log\{1 + \exp(\boldsymbol{\beta}^{\text{T}}\boldsymbol{x}_i^*)\}}{\pi_i^*}.$$

---

cess every observation in the full dataset when calculating the approximately optimal subsampling probabilities, making the computational time at the order of N.

For faster calculation, the variance-covariance matrix of $\tilde{\boldsymbol{\beta}}^{\text{OS}}$ can be estimated by

$$\tilde{\boldsymbol{V}}^{\text{OS}} = (\boldsymbol{M}_L^*)^{-1}\boldsymbol{V}_{Nc}^*(\boldsymbol{M}_L^*)^{-1}, \tag{6}$$

where

$$\boldsymbol{M}_L^* = \frac{1}{(n_0+n_1)N}\sum_{i=1}^{n_0+n_1} \frac{p(\boldsymbol{x}_i^*, \tilde{\boldsymbol{\beta}}^{\text{OS}})\{1 - p(\boldsymbol{x}_i^*, \tilde{\boldsymbol{\beta}}^{\text{OS}})\}\boldsymbol{x}_i^*(\boldsymbol{x}_i^*)^{\text{T}}}{\pi_i^*}, \quad \text{and}$$

$$\boldsymbol{V}_{Nc}^* = \frac{1}{(n_0+n_1)^2N^2}\sum_{i=1}^{n_0+n_1} \frac{\{y_i^* - p(\boldsymbol{x}_i^*, \tilde{\boldsymbol{\beta}}^{\text{OS}})\}^2\boldsymbol{x}_i^*(\boldsymbol{x}_i^*)^{\text{T}}}{(\pi_i^*)^2}.$$

Note that the Algorithm 2 is built under the circumstance that the regression model is correctly specified. Thus the model selection or the covariate transformation should be done in advance. Another thing is that, when practically implementing Algorithm 2, the second stage sample size should be always much larger than pilot sample size. This is a theoretical assumption ensuring the asymptotic normality of $\tilde{\boldsymbol{\beta}}^{\mathrm{OS}}$, and guarantees the contribution of the second stage sample to the final estimator far beyond the first stage sample. These two statements are applicable to all optimal subsampling methods in this article.

2.1 Optimal subsampling method for softmax regression

The OSMAC was investigated to include softmax regression, which is also called multinomial logistic regression, in Yao and Wang (2018). Suppose that the response of the softmax regression, which is a categorical variable, contains $K+1$ distinct outcomes, say $y_i \in \{0, 1, ..., K\}$. The softmax regression has the following form

$$P(y_i = k|\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_k)}{\sum_{j=0}^{K}\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_j)}, \quad k = 0, 1, ..., K, \tag{7}$$

where $\boldsymbol{\beta}_k$ is the unknown coefficient for category k. Let $\boldsymbol{\beta}_0 = \mathbf{0}$ for identifiability. The unknown parameter for the whole model is denoted as $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}}, ..., \boldsymbol{\beta}_K^{\mathrm{T}})^{\mathrm{T}}$, and (7) becomes

$$\begin{aligned} P(y_i = 0|\boldsymbol{x}_i) = p_0(\boldsymbol{x}_i|\boldsymbol{\beta}) &= \frac{1}{1+\sum_{j=1}^{K}\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_j)}, \\ P(y_i = k|\boldsymbol{x}_i) = p_k(\boldsymbol{x}_i|\boldsymbol{\beta}) &= \frac{\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_k)}{1+\sum_{j=1}^{K}\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_j)}. \end{aligned}$$

Under this model, the log-likelihood function for the observed dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ is

$$\ell_{\mathrm{so}}(\boldsymbol{\beta}) = \sum_{i=1}^{N}\left[\sum_{k=1}^{K} I(y_i = k)\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} - \log\left\{1 + \sum_{j=1}^{K}\exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}_j)\right\}\right].$$

Maximizing this log-likelihood function, we can obtain the full data MLE $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ though Newon-Raphson method. By deriving the variance-covariance matrix of a general subsample estimator for softmax regression, the optimal subsampling probabilities are

$$\pi_{\mathrm{so},i}^{\mathrm{optA}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{\|\boldsymbol{M}_S^{-1}\{\boldsymbol{s}_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \otimes \boldsymbol{x}_i\}\|}{\sum_{j=1}^N \|\boldsymbol{M}_S^{-1}\{\boldsymbol{s}_j(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \otimes \boldsymbol{x}_j\}\|}, \quad \text{under A-optimality criterion, and} \tag{8}$$

$$\pi_{\mathrm{so},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{\|\boldsymbol{s}_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\|\|\boldsymbol{x}_i\|}{\sum_{j=1}^N \|\boldsymbol{s}_j(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\|\|\boldsymbol{x}_j\|}, \quad \text{under L-optimality criterion,} \tag{9}$$

where $\boldsymbol{M}_S = N^{-1}\sum_{i=1}^N \boldsymbol{\Psi}_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \otimes (\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}})$; $\boldsymbol{\Psi}_i(\boldsymbol{\beta})$ is a $K \times K$ matrix whose k-th diagonal element is $\boldsymbol{\Psi}_{i,(k,k)}(\boldsymbol{\beta}) = p_k(\boldsymbol{x}_i, \boldsymbol{\beta}) - p_k^2(\boldsymbol{x}_i, \boldsymbol{\beta})$ and k_1k_2-th off-diagonal element is $\boldsymbol{\Psi}_{i,(k_1,k_2)}(\boldsymbol{\beta}) = -p_{k_1}(\boldsymbol{x}_i, \boldsymbol{\beta})p_{k_2}(\boldsymbol{x}_i, \boldsymbol{\beta})$; and $s_i(\boldsymbol{\beta}) \in \mathcal{R}^K$ with k-th element being $\boldsymbol{s}_{i,k}(\boldsymbol{\beta}) = I(y_i = k) - p_i(k, \boldsymbol{\beta})$. With the strategy that uses pilot sample estimator to replace $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ when calculating optimal subsampling probabilities, we have the adaptive optimal subsampling algorithm for softmax regression.

2.2 Income dataset

The behavior of Algorithm 2 is illustrated by the income dataset (Dua and Graff, 2017), which contains 48842 observations in total. The response is an indicator variable which shows whether one person's income is over 50K or not, and around 24% of participants have income exceeding 50K. We use 5 continuous covariates to build the logistic model, which are age, final weight (fnlwgt), education (edu), capital loss (loss) and working hours per week (hours). The original dataset was partitioned into training dataset and test dataset. We combined these two datasets, selected variables involving in the logistic model, and name this newly generated data as `adult1`. Applying `glm` function in `stats` package (R Core Team, 2020) to `adult1`, we can obtain the coefficient estimator for the covariates using the following chunk of code.

```
adult <- read.table("Code/adult.data", sep = ",")
```

```
test <- read.table("Code/adult.test", sep = ",", skip = 1)
test$V15 <- gsub("\\.", "", test$V15)
adult <- rbind(adult, test)
adult1 <- subset(adult, select = c("V1", # age
                                   "V3", # fnlwgt
                                   "V5", # edu
                                   "V12", # loss
                                   "V13", # hours
                                   "V15", # income
                                   NULL))
adult1$V15 <- as.numeric(adult1$V15 == " >50K")
income.glm <- glm(V15 ~ ., data = adult1, family = "binomial")
summary(income.glm)

##
## Call:
## glm(formula = V15 ~ ., family = "binomial", data = adult1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0587  -0.6890  -0.4364  -0.1376   3.0926
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.587e+00  9.436e-02 -91.009  < 2e-16 ***
## V1           4.594e-02  9.518e-04  48.266  < 2e-16 ***
## V3           6.007e-07  1.148e-07   5.231 1.68e-07 ***
## V5           3.410e-01  5.315e-03  64.156  < 2e-16 ***
## V12          5.616e-04  2.643e-05  21.244  < 2e-16 ***
## V13          4.202e-02  1.033e-03  40.669  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 53751  on 48841  degrees of freedom
## Residual deviance: 42995  on 48836  degrees of freedom
## AIC: 43007
##
## Number of Fisher Scoring iterations: 5
```

It is seen that every covariates is statistically significant, and as any covariate increases, the probability for a person with an income larger than 50K increases.

In the following, we implemented Algorithm 2 to `adult1` by function `AdpOptSubLog`, in which the subsample estimator is calculated through `svyglm` function from `survey` package (Lumley, 2020) along with `weights` option.

```
X <- cbind(1, as.matrix(adult1[, -dim(adult1)[2]]))
y <- adult1$V15
set.seed(123)
AdpOptSubLog(X, y, r0 = 500, r = 1000, optmethod = "A", data = adult1,
             covariate = "V1 + V3 + V5 + V12 + V13")

##           coefficients       stdErr     Zvalue        Pvalue
## intercept -8.791713e+00 3.895829e-01 -22.566987 9.147120e-113
## beta1      4.269838e-02 5.432964e-03   7.859132  3.868057e-15
## beta2      1.528809e-06 5.649197e-07   2.706241  6.804961e-03
## beta3      3.535131e-01 2.817009e-02  12.549236  4.013658e-36
## beta4      8.640746e-04 1.386894e-04   6.230288  4.655796e-10
## beta5      4.168246e-02 5.534931e-03   7.530801  5.043002e-14
```

In the function `AdpOptSubLog`, `X` is the covariate matrix, `y` is the response variable with numerical format, `r0` stands for the pilot sample size, `r` stands for the second step sample size, and `optmethod` indicates the optimality criterion, which can be "A" and "L". The output gives coefficient estimators and estimated standard errors, along with the z statistics and p values used to test whether the MLE for the corresponding covariate equals to 0 or not. For an arbitrary β_j, the z statistic is calculated by

$$z_j = \frac{\tilde{\beta}_j^{\mathrm{OS}}}{\sqrt{\tilde{V}_{jj}^{\mathrm{OS}}}},$$

where $\sqrt{\tilde{V}_{jj}^{\mathrm{OS}}}$ is the estimated standard error and the estimated standard error is the squared root of j-th diagonal element of $\tilde{\boldsymbol{V}}^{\mathrm{OS}}$ in (6).

3 Efficient optimal subsampling for logistic regression

In this section, we introduce two approaches proposed by Wang (2019b) to improve the OSMAC, where the first one is to use unweighted subsample estimators and the other one is to adopt Poisson subsampling.

3.1 Efficient optimal subsampling with unweighted estimator

In Algorithm 2, $\tilde{\boldsymbol{\beta}}^{\mathrm{OS}}$ is obtained by maximizing weighted target function because the expression of the optimal subsampling probabilities involves y_i. From (1), we can see that data points with higher subsampling probabilities contribute relatively less towards the weighted target function. Note that the higher subsampling probability one data point has, the more information that observation carries. Thus, the weighted target function cannot utilize the information of a sample as efficient as an unweighted target function. Given a subsample $\{\boldsymbol{x}_i^*, y_i^*\}_{i=1}^n$, the general unweighted subsample estimator $\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{sub}}$ proposed by Wang (2019b) is obtained by maximizing

$$\ell_{\mathrm{uw}}^*(\boldsymbol{\beta}) = \sum_{i=1}^n \left[y_i^* \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}_i^* - \log\{1 + \exp(\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}_i^*)\} \right].$$

The $\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{sub}}$ is biased and a bias correction procedure is needed. Algorithm 3 summarizes how to implement unweighted estimator in the optimal subsampling method and how to correct the bias.

3.2 Efficient optimal subsampling using Poisson subsampling

Besides subsampling with replacement, Poisson subsampling was considered in Wang (2019b). For Poisson subsampling, each observation is assigned to a subsampling probability and we decide to include a data point into a sample by conducting a Bernoulli trail with the assigned subsampling probability as the successful rate. The observations in the subsample drawn by Poisson subsampling can be independent to each other unconditionally to the full data. That means, we can calculate the subsampling probabilities for i-th observation and decide whether to include i-th observation into subsample only based on the information of the i-th data point. Whereas for the subsampling with replacement, we have to draw a large indexes of samples from N numbers with pre-specified subsampling probabilities. For enormously large N such that N exceeding the memory limit of the computer, the subsampling with replacement fails to be applied. Another advantage of Poisson subsampling is that no replicate observation exists in the subsample. Furthermore, the sample size is a random variable for Poisson subsampling, and we need to use

Algorithm 3 Efficient adaptive optimal subsampling algorithm

Pilot sampling:

- Assign subsampling probabilities $\pi_i^0 = c_0^{1-y_i} c_1^{y_i}$ to each data point, where $c_0 = c_1 = \frac{1}{N}$ for uniform subsampling and $c_0 = 1/(2N - 2\sum_{i=1}^N y_i)$, $c_1 = 1/(2\sum_{i=1}^N y_i)$ for case control subsampling.
- Draw n_0 data points with replacement based on $\{\pi_i^0\}_{i=1}^N$ and denote the sampled dataset as $\{\boldsymbol{x}_i^{*0}, y_i^{*0}\}_{i=1}^{n_0}$.

Estimation for pilot sampling:

- Obtain the unweighted estimator $\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}$ by maximizing

$$\ell_{\text{uw}}^{*0}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0} \left[y_i^{*0} \boldsymbol{\beta}^{\text{T}} \boldsymbol{x}_i^{*0} - \log\{1 + \exp(\boldsymbol{\beta}^{\text{T}} \boldsymbol{x}_i^{*0})\} \right].$$

- Correct bias and the pilot sample estimator is $\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0} = \hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0} + (\log(c_0/c_1), \underbrace{0, ..., 0}_{d-1})^{\text{T}}$.

Second step sampling:

- Calculate the approximate optimal subsampling probabilities $\{\tilde{\pi}_i\}_{i=1}^N$ based on (4) or (5) with $\hat{\boldsymbol{\beta}}^{\text{sub},0}$ being substituted by $\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}$.
- Sample n_1 data points with replacement based on $\{\tilde{\pi}_i\}_{i=1}^N$ and denote the sampled dataset as $\{\boldsymbol{x}_i^{*1}, y_i^{*1}\}_{i=1}^{n_1}$.

Estimation for second step sampling:

- Obtain the unweighted estimator $\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1}$ for second step sample by maximizing

$$\ell_{\text{uw}}^{*1}(\boldsymbol{\beta}) = \sum_{i=1}^{n_1} \left[y_i^{*1} \boldsymbol{\beta}^{\text{T}} \boldsymbol{x}_i^{*1} - \log\{1 + \exp(\boldsymbol{\beta}^{\text{T}} \boldsymbol{x}_i^{*1})\} \right].$$

- The second step estimator is obtained by correcting bias, say $\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1} = \hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1} + \tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}$.

Combination: The final estimator $\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub}}$ is obtained by

$$\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub}} = \left\{ \ddot{\ell}_{\text{uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}) + \ddot{\ell}_{\text{uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1}) \right\}^{-1} \left\{ \ddot{\ell}_{\text{uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}) \tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0} + \ddot{\ell}_{\text{uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1}) \tilde{\boldsymbol{\beta}}^{\text{sub},1} \right\},$$

where

$$\ddot{\ell}_{\text{uw}}^{*0}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0} p(\boldsymbol{x}_i^{*0}, \boldsymbol{\beta})\{1 - p(\boldsymbol{x}_i^{*0}, \boldsymbol{\beta})\} \boldsymbol{x}_i^{*0} (\boldsymbol{x}_i^{*0})^{\text{T}};$$

$$\ddot{\ell}_{\text{uw}}^{*1}(\boldsymbol{\beta}) = \sum_{i=1}^{n_1} p(\boldsymbol{x}_i^{*1}, \boldsymbol{\beta})\{1 - p(\boldsymbol{x}_i^{*1}, \boldsymbol{\beta})\} \boldsymbol{x}_i^{*1} (\boldsymbol{x}_i^{*1})^{\text{T}}.$$

The variance-covariance matrix of $\tilde{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub}}$ can be estimated by

$$\begin{aligned}\tilde{\boldsymbol{V}}_{\text{uw}}^{\text{sub}} = &\left\{ \ddot{\ell}_{\text{uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}) + \ddot{\ell}_{\text{uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1}) \right\}^{-1} \left[\sum_{i=1}^{n_0} \{y_i^{*0} - p(\boldsymbol{x}_i^{*0}, \hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0})\}^2 \boldsymbol{x}_i^{*0} (\boldsymbol{x}_i^{*0})^{\text{T}} \right. \\ &\left. + \sum_{i=1}^{n_1} \{y_i^{*1} - p(\boldsymbol{x}_i^{*1}, \hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1})\}^2 \boldsymbol{x}_i^{*1} (\boldsymbol{x}_i^{*1})^{\text{T}} \right] \left\{ \ddot{\ell}_{\text{uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},0}) + \ddot{\ell}_{\text{uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\text{uw}}^{\text{sub},1}) \right\}^{-1}. \quad (10)\end{aligned}$$

the expected sample size to control it. The procedure of a general Poisson subsampling is described in Algorithm 4.

Algorithm 4 Poisson subsampling

Input: $\{\boldsymbol{x}_i, y_i, \pi_i\}_{i=1}^N$, n is the expected sample size, $\pi_i \le 1/n$
Output: Sample set $\mathcal{S}$
Initialization: $\mathcal{S} \leftarrow \varnothing$
for i in $\{1, 2, ..., N\}$ **do**
  $u \sim \text{Unif}(0, 1)$,
  if $u < n\pi_i$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup (\boldsymbol{x}_i, y_i, \pi_i)$,
  end if
end for

To keep all those features of Poisson subsampling, given a pilot sample with corresponding subsampling probabilities $\{\boldsymbol{x}_i^{*0}, y_i^{*0}, \pi_i^{*0}\}_{i=1}^{n_0^*}$ and the pilot coefficient estimator $\tilde{\boldsymbol{\beta}}^{\text{ps},0}$, the approximated optimal subsampling probabilities under A-optimality and L-optimality criteria are

$$\pi_{\text{ps},i}^{\text{optA}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0}) = \frac{|y_i - p(\boldsymbol{x}_i, \tilde{\boldsymbol{\beta}}^{\text{ps},0})|\|\boldsymbol{M}_P^{-1}(\tilde{\boldsymbol{\beta}}^{\text{ps},0})\boldsymbol{x}_i\|}{\phi^{\text{optA}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0})}, \quad i = 1, ..., N, \quad \text{and} \tag{11}$$

$$\pi_{\text{ps},i}^{\text{optL}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0}) = \frac{|y_i - p(\boldsymbol{x}_i, \tilde{\boldsymbol{\beta}}^{\text{ps},0})|\|\boldsymbol{x}_i\|}{\phi^{\text{optL}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0})}, \quad i = 1, ..., N, \quad \text{respectively,} \tag{12}$$

where

$$\phi^{\text{optA}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0}) = \sum_{j=1}^{n_0^*} \frac{|y_j^{*0} - p(\boldsymbol{x}_j^{*0}, \tilde{\boldsymbol{\beta}}^{\text{ps},0})|\|\boldsymbol{M}_P^{-1}(\tilde{\boldsymbol{\beta}}^{\text{ps},0})\boldsymbol{x}_j^{*0}\|}{(n_0\pi_j^{*0}) \wedge 1},$$

$$\phi^{\text{optL}}(\tilde{\boldsymbol{\beta}}^{\text{ps},0}) = \sum_{j=1}^{n_0^*} \frac{|y_j^{*0} - p(\boldsymbol{x}_j^{*0}, \tilde{\boldsymbol{\beta}}^{\text{ps},0})|\|\boldsymbol{x}_j\|}{(n_0\pi_j^{*0}) \wedge 1},$$

$$\boldsymbol{M}_P(\tilde{\boldsymbol{\beta}}^{\text{ps},0}) = \frac{1}{N} \sum_{i=1}^{n_0^*} \frac{p(\boldsymbol{x}_i^{*0}, \tilde{\boldsymbol{\beta}}^{\text{ps},0})\{1 - p(\boldsymbol{x}_i^{*0}, \tilde{\boldsymbol{\beta}}^{\text{ps},0})\}\boldsymbol{x}_i^{*0}(\boldsymbol{x}_i^{*0})^{\text{T}}}{(n_0\pi_j^{*0}) \wedge 1},$$

and n_1 is the expected sample size. The adaptive optimal subsampling methods with Poisson subsampling is described in Algorithm 5.

3.3 Income dataset

Algorithm 3 is realized by function `AdpOptUWLog`. The following code applies the function `AdpOptUWLog` to the income dataset. The standard errors are calculated from (10).

Algorithm 5 Efficient adaptive optimal subsampling algorithm using Poisson subsampling

Pilot sampling:

– Run Algorithm 4 with expected sample size n_0 and subsampling probabilities π_i^0.

– Obtain a pilot sample with sample size n_0^*, say $\{\boldsymbol{x}_i^{*0}, y_i^{*0}, \pi_i^{*0}\}_{i=1}^{n_0^*}$.

Estimation for pilot sampling:

– Obtain $\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0}$ by maximizing

$$\ell_{\mathrm{ps,uw}}^{*0}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0^*} (n_0\pi_i^{*0} \vee 1)\left[y_i^{*0}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^{*0} - \log\{1+\exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^{*0})\}\right].$$

– Correct bias and the pilot sample estimator is $\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0} = \hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0} + (\log(c_0/c_1), \underbrace{0, ..., 0}_{d-1})^{\mathrm{T}}$.

Second step sampling:

– Calculate the approximate optimal subsampling probabilities $\{\pi_{\mathrm{ps},i}^{\mathrm{optA}}(\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\}_{i=1}^N$ or $\{\pi_{\mathrm{ps},i}^{\mathrm{optL}}(\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\}_{i=1}^N$ based on (11) or (12).

– Run Algorithm 4 with expected sample size n_1 and subsampling probabilities $\{\pi_{\mathrm{ps},i}^{\mathrm{optA}}(\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\}_{i=1}^N$ or $\{\pi_{\mathrm{ps},i}^{\mathrm{optL}}(\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\}_{i=1}^N$ to obtain the second step sample, denoted as $\{\boldsymbol{x}_i^{*1}, y_i^{*1}, \pi_i^{*1}\}_{i=1}^{n_1^*}$, where n_1^* is the true sample size.

Estimation for second step sampling:

– Obtain $\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1}$ for second step sample by maximizing

$$\ell_{\mathrm{ps,uw}}^{*1}(\boldsymbol{\beta}) = \sum_{i=1}^{n_1^*} (n_1\pi_i^{*1} \vee 1)\left[y_i^{*1}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^{*1} - \log\{1+\exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^{*1})\}\right].$$

– The second step estimator can be obtained by correcting bias, say $\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1} = \hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1} + \tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0}$.

Combination: The final estimator $\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps}}$ is obtained by

$$\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps}} = \left\{\ddot{\ell}_{\mathrm{ps,uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0}) + \ddot{\ell}_{\mathrm{ps,uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1})\right\}^{-1}\left\{\ddot{\ell}_{\mathrm{ps,uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0} + \ddot{\ell}_{\mathrm{ps,uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1})\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1}\right\},$$

where

$$\ddot{\ell}_{\mathrm{ps,uw}}^{*0}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0^*} p(\boldsymbol{x}_i^{*0}, \boldsymbol{\beta})\{1-p(\boldsymbol{x}_i^{*0}, \boldsymbol{\beta})\}\boldsymbol{x}_i^{*0}(\boldsymbol{x}_i^{*0})^{\mathrm{T}};$$
$$\ddot{\ell}_{\mathrm{ps,uw}}^{*1}(\boldsymbol{\beta}) = \sum_{i=1}^{n_1^*} p(\boldsymbol{x}_i^{*1}, \boldsymbol{\beta})\{1-p(\boldsymbol{x}_i^{*1}, \boldsymbol{\beta})\}\boldsymbol{x}_i^{*1}(\boldsymbol{x}_i^{*1})^{\mathrm{T}}.$$

The variance-covariance matrix of $\tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps}}$ can be estimated by

$$\tilde{\boldsymbol{V}}_{\mathrm{uw}}^{\mathrm{ps}} = \left\{\ddot{\ell}_{\mathrm{ps,uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0}) + \ddot{\ell}_{\mathrm{ps,uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1})\right\}^{-1}\left[\sum_{i=1}^{n_0^*}\{y_i^{*0} - p(\boldsymbol{x}_i^{*0}, \hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})\}^2\boldsymbol{x}_i^{*0}(\boldsymbol{x}_i^{*0})^{\mathrm{T}}\right.$$
$$\left. + \sum_{i=1}^{n_1^*}\{y_i^{*1} - p(\boldsymbol{x}_i^{*1}, \hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1})\}^2\boldsymbol{x}_i^{*1}(\boldsymbol{x}_i^{*1})^{\mathrm{T}}\right]\left\{\ddot{\ell}_{\mathrm{ps,uw}}^{*0}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0}) + \ddot{\ell}_{\mathrm{ps,uw}}^{*1}(\hat{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},1})\right\}^{-1}. \quad (13)$$

```
AdpOptUWLog(X, y, r0 = 500, r = 1000, optmethod = "A", data = adult1,
            covariate = "V1 + V3 + V5 + V12 + V13")

##             coefficients       stdErr     Zvalue        Pvalue
## intercept -8.474380e+00 3.510944e-01 -24.137046 1.021383e-128
## beta1      4.719963e-02 4.355923e-03  10.835736  2.330840e-27
## beta2      5.618032e-07 4.707567e-07   1.193405  2.327110e-01
## beta3      3.268313e-01 2.371188e-02  13.783442  3.206122e-43
## beta4      8.420474e-04 1.227689e-04   6.858799  6.944199e-12
## beta5      3.945990e-02 4.427490e-03   8.912477  4.990482e-19
```

Function `AdpOptPosLog` is coded according to Algorithm 5, and apply this function to the income dataset using the code below.

```
AdpOptPosLog(X, y, r0 = 500, r = 1000, optmethod = "A", data = adult1,
             covariate = "V1 + V3 + V5 + V12 + V13")

## [[1]]
##             coefficients       stdErr      Zvalue        Pvalue
## intercept -8.662216e+00 3.204785e-01 -27.0290055 6.743402e-161
## beta1      5.055410e-02 4.467591e-03  11.3157404  1.096725e-29
## beta2      4.838215e-07 5.175832e-07   0.9347705  3.499066e-01
## beta3      3.597660e-01 2.232577e-02  16.1143842  2.021679e-58
## beta4      7.112636e-04 1.103245e-04   6.4470131  1.140759e-10
## beta5      3.299479e-02 3.512693e-03   9.3930180  5.830516e-21
##
## [[2]]
##   pilot.sample.size second.sample.size
## 1               493               1107
```

Because the sample size for Poisson sampling is random, we record the true sample size in both stages. In `AdpOptPosLog`, `r0` and `r`, which are the expected pilot sample size and expected second stage sample size, respectively, are set to be 500 and 1000. We can see, in this example, the true pilot sample size is 493 and true second stage sample size is 1107. The `optmethod` could be "A", "L" and "LCC", where "LCC" represents the local case control sampling introduced in Fithian and Hastie (2014), and the second step estimator is used as the final estimator. When selecting `optmethod = "LCC"`, `r` is not meaningful since the subsampling probabilities at second stage become $|y_i - p(\boldsymbol{x}_i, \tilde{\boldsymbol{\beta}}_{\mathrm{uw}}^{\mathrm{ps},0})|$. The expected sample size is determined by the discrepancy between the real value and estimated probabilities, and is at the same order of N.

Table 1: MSE, averaged second step sample size and running time of different methods for the income data when $n_0 = 500$ and $n_1 = 1000$ are fixed for 1000 replications. The sample size for uniform subsampling is $n_0 + n_1$ for fair comparison. A.S. Sample Size means the averaged second step sample size used for each algorithm.

Method	MSE	A.S. Sample Size	CPU Seconds
Algorithm 2 optA	0.170	1000	41.598
Algorithm 2 optL	0.271	1000	39.590
Algorithm 3 optA	0.127	1000	33.067
Algorithm 3 optL	0.271	1000	31.369
Algorithm 5 optA	0.106	1041.478	38.790
Algorithm 5 optL	0.238	1020.293	36.681
LCC	0.0176	13824.657	153.129
Uniform	0.317	NA	6.860
Full data CPU seconds: 230.116			

Table 1 compares the statistical efficiency and computing efficiency of the proposed algorithms with uniform subsampling and local case control sampling for the income dataset. The statistical efficiency is measured by MSE, where MSE is calculated by $S^{-1}\sum_{i=1}^{S}\|\tilde{\boldsymbol{\beta}}_i^* - \hat{\boldsymbol{\beta}}_{\text{MLE}}\|^2$ with S being the number of replications and $\tilde{\boldsymbol{\beta}}_i^*$ being the final estimator of the targeted algorithm for i-th replication. All computations are processed on a MacBook Pro with a 2.5 GHz Intel Core i7 processor and 16 GB memory. Table 1 shows that the uniform subsampling takes the least time since only one sampling step is involved and no need to compute the subsampling probabilities. The performances of all proposed algorithms in estimation efficiency are better than the uniform subsampling. Among these approaches, the local case control sampling is the most efficient one in approximating $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ because this method draws greatly more second step samples than others. As a consequence, the local case control sampling has a heavier computational burden than the proposed algorithms. Obviously, directly calculating $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ with the full data is the most time consuming method. For the statistical efficiency of these three proposed algorithms, Algorithm 5 outperforms the other two, and Algorithm 2 is the least efficient one, indicating that using unweighted estimator and Poisson subsampling helps improve the estimation accuracy. In addition, it can be seen that Algorithms under L-optimality are less efficient in coefficient estimation but more efficient in terms of computing time than Algorithms under A-optimality.

4 Optimal subsampling method for generalized linear models

Consider a generalized linear model with expression

$$f(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\beta}) = h(y_i)\exp\left[y_i g(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}) - c\{g(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})\}\right], \tag{14}$$

where $h(\cdot)$, $g(\cdot)$ and $c(\cdot)$ are known functions. The $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ can be obtained by maximizing

$$\ell_{\mathrm{glm}}(\boldsymbol{\beta}) = \sum_{i=1}^{N}\log f(y_i|\boldsymbol{x}_i,\boldsymbol{\beta})$$

through the Newton-Raphson method, which can be achieved in $O(\eta N d^2)$ time, where η is the number of iterations for the Newton-Raphson method to converge. Assign subsampling probabilities to each observation. Draw n observations with replacement and denote them as $\{\boldsymbol{x}_i^*, y_i^*, \pi_i^*\}_{i=1}^n$. The subsample estimator $\hat{\boldsymbol{\beta}}_{\mathrm{sub}}^{\mathrm{glm}}$ is obtained by maximizing the weighted target function

$$\ell_{\mathrm{glm}}^*(\boldsymbol{\beta}) = \sum_{i=1}^{n}\frac{\log f(y_i^*|\boldsymbol{x}_i^*,\boldsymbol{\beta})}{\pi_i^*}. \tag{15}$$

By minimizing the asymptotic MSE of $\hat{\boldsymbol{\beta}}_{\mathrm{sub}}^{\mathrm{glm}}$, the optimal subsampling probabilities under A-optimality criterion are

$$\pi_{\mathrm{glm},i}^{\mathrm{optA}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}|\|\boldsymbol{M}_G^{-1}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\dot{g}(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_i\|}{\sum_{j=1}^{N}|y_j - \dot{c}\{g(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}|\|\boldsymbol{M}_G^{-1}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\dot{g}(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_j\|}, \tag{16}$$

where $\dot{c}(\cdot)$ and $\dot{g}(\cdot)$ are the first-order derivatives of $c(\cdot)$ and $g(\cdot)$; and

$$\boldsymbol{M}_G(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\ddot{g}(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}[\dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\} - y_i] + \ddot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}\dot{g}^2(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right\},$$

with $\ddot{c}(\cdot)$ and $\ddot{g}(\cdot)$ being the second-order derivatives of $c(\cdot)$ and $g(\cdot)$. The optimal subsampling probabilities under L-optimality criterion are

$$\pi_{\mathrm{glm},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}|\|\dot{g}(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_i\|}{\sum_{j=1}^{N}|y_j - \dot{c}\{g(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}|\|\dot{g}(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\boldsymbol{x}_j\|}. \tag{17}$$

We need $O(Nd^2)$ time to compute $\pi_{\mathrm{glm},i}^{\mathrm{optA}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})$ and $O(Nd)$ time to compute $\pi_{\mathrm{glm},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})$. From (15), we can see that the weighted target function is easily inflated by extreme small subsampling probabilities. To solve this, the authors in Ai et al. (2019) used a threshold to constraint the value of $|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})\}|$ from below. In such way, given a pilot sample estimator $\hat{\boldsymbol{\beta}}^{\mathrm{glm},0}$ and a pre-specified threshold δ, the approximated optimal subsampling probabilities are

$$\hat{\pi}_{\mathrm{glm},i}^{\mathrm{optA}}(\hat{\boldsymbol{\beta}}^{\mathrm{glm},0}) = \frac{\max\{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\}|,\delta\}\|\boldsymbol{M}_G^{-1}(\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\dot{g}(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\boldsymbol{x}_i\|}{\sum_{j=1}^{N}\max\{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\}|,\delta\}\|\boldsymbol{M}_G^{-1}(\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\dot{g}(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\boldsymbol{x}_j\|} \tag{18}$$

under A-optimality criterion and

$$\hat{\pi}_{\mathrm{glm},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}^{\mathrm{glm},0}) = \frac{\max\{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\}|,\delta\}\|\dot{g}(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\boldsymbol{x}_i\|}{\sum_{j=1}^{N}\max\{|y_i - \dot{c}\{g(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\}|,\delta\}\|\dot{g}(\boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\boldsymbol{x}_j\|} \tag{19}$$

under L-optimality criterion. The adaptive optimal subsampling algorithm for generalized linear regression is summarized in Algorithm 6. It has been proved in Ai et al. (2019) that the resultant estimator of Algorithm 6 is asymptotically normal and the rate of convergence is $O(n_1^{-1/2})$ under some mild assumptions.

4.1 Poisson Regression

Poisson regression is widely used for modeling count data, and is one of the generalized linear models. Under (14), the poisson regression has $h(y_i) = 1/(y_i!)$, $g(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ and $c(\cdot) = \exp(\cdot)$, and is of the form

$$f(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\beta}) = \frac{1}{y_i!}\exp\left\{y_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} - \exp(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})\right\}. \tag{21}$$

Algorithm 6 Adaptive optimal subsampling algorithm for generalized linear models

---

Pilot sampling:

- Assign $\pi_i^0 = N^{-1}$ to each observation.
- Choose n_0 data points with replacement and record the subsample as $\{\boldsymbol{x}_i^{*0}, y^{*0}, \pi_i^{*0}\}_{i=1}^{n_0}$.
- Obtain the pilot sample estimator $\hat{\boldsymbol{\beta}}^{\text{glm},0}$ by maximizing

$$\ell_{\text{glm}}^{*0}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0} \frac{\log f(y_i^{*0}|\boldsymbol{x}_i^{*0}, \boldsymbol{\beta})}{\pi_i^{*0}}.$$

Second step sampling:

- Calculate the approximate optimal subsampling probabilities $\{\hat{\pi}_{\text{glm},i}^{\text{optA}}(\hat{\boldsymbol{\beta}}^{\text{glm},0})\}_{i=1}^N$ or $\hat{\pi}_{\text{glm},i}^{\text{optL}}(\hat{\boldsymbol{\beta}}^{\text{glm},0})$ based on (18) or (19).
- Draw n_1 samples with replacement based on those approximate optimal subsampling probabilities.
- Record the second step subsample and the corresponding subsampling probabilities $\{\boldsymbol{x}_i^{*1}, y^{*1}, \pi_i^{*1}\}_{i=1}^{n_1}$.

Estimation: Combine the pilot sample and second stage sample and denote it as $\{\boldsymbol{x}_i^{*}, y^{*}, \pi_i^{*}\}_{i=1}^{n_0+n_1}$. Obtain the final estimator $\tilde{\boldsymbol{\beta}}_{\text{glm}}$ by maximizing

$$\ell_{\text{glm}}^{*}(\boldsymbol{\beta}) = \sum_{i=1}^{n_0+n_1} \frac{\log f(y_i^{*}|\boldsymbol{x}_i^{*}, \boldsymbol{\beta})}{\pi_i^{*}}.$$

Estimate the variance-covariance matrix of $\tilde{\boldsymbol{\beta}}_{\text{glm}}$ by

$$\tilde{\boldsymbol{V}} = (\boldsymbol{M}_G^*)^{-1}\boldsymbol{V}_G^*(\boldsymbol{M}_G^*)^{-1}, \tag{20}$$

where

$$\boldsymbol{M}_G^* = \sum_{i=1}^{n_0+n_1} \frac{\ddot{g}(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\boldsymbol{x}_i^*(\boldsymbol{x}_i^*)^{\text{T}}[\dot{c}\{g(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\} - y_i^*] + \ddot{c}\{g(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\}\dot{g}^2(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\boldsymbol{x}_i^*(\boldsymbol{x}_i^*)^{\text{T}}}{(n_0+n_1)N\pi_i^*},$$

$$\boldsymbol{V}_G^* = \sum_{i=1}^{n_0+n_1} \frac{[y_i^* - \dot{c}\{g(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\}]^2\dot{g}^2(\tilde{\boldsymbol{\beta}}_{\text{glm}}^{\text{T}}\boldsymbol{x}_i^*)\boldsymbol{x}_i^*(\boldsymbol{x}_i^*)^{\text{T}}}{(n_0+n_1)^2N^2(\pi_i^*)^2}.$$

Given a prior estimator $\hat{\boldsymbol{\beta}}^{\text{glm},0}$, the approximated optimal subsampling probabilities in (18) and (19) become

$$\hat{\pi}_{\text{pr},i}^{\text{optA}}(\hat{\boldsymbol{\beta}}^{\text{glm},0}) = \frac{\max\{|y_i - \exp(\boldsymbol{x}_i^{\text{T}}\hat{\boldsymbol{\beta}}^{\text{glm},0})|, \delta\}\|\boldsymbol{M}_P^{-1}(\hat{\boldsymbol{\beta}}^{\text{glm},0})\boldsymbol{x}_i\|}{\sum_{j=1}^N \max\{|y_i - \exp(\boldsymbol{x}_i^{\text{T}}\hat{\boldsymbol{\beta}}^{\text{glm},0})|, \delta\}\|\boldsymbol{M}_P^{-1}(\hat{\boldsymbol{\beta}}^{\text{glm},0})\boldsymbol{x}_j\|} \quad \text{and} \tag{22}$$

$$\hat{\pi}_{\text{pr},i}^{\text{optL}}(\hat{\boldsymbol{\beta}}^{\text{glm},0}) = \frac{\max\{|y_i - \exp(\boldsymbol{x}_i^{\text{T}}\hat{\boldsymbol{\beta}}^{\text{glm},0})|, \delta\}\|\boldsymbol{x}_i\|}{\sum_{j=1}^N \max\{|y_i - \exp(\boldsymbol{x}_i^{\text{T}}\hat{\boldsymbol{\beta}}^{\text{glm},0})|, \delta\}\|\boldsymbol{x}_j\|}, \quad \text{respectively,} \tag{23}$$

where $\boldsymbol{M}_P = \frac{1}{N}\sum_{i=1}^{N}\exp(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$. Plug in (21), (22) and (23) into Algorithm 6, and we can have the adaptive optimal subsampling algorithm for poisson regression.

4.2 Bike sharing dataset

The bike sharing dataset, which models the number of bikes rented hourly under different conditions, is used to demonstrate the effectiveness of the Algorithm 6 to the poisson regression. This dataset contains 17379 observations, and 4 covariates are included to the model, consisting of a binary variable "workingday" to indicate whether a certain day is a working day or not, 3 continuous variables which are "temp" (temperature), "hum" (humidity) and "windspeed" (windspeed). The organized dataset is named `hour1` and the coefficient estimator for `hour1` is computed by `glm` using `family = "poisson"`. The following code shows how to obtain the MLE for the full dataset.

```
hour <- read.csv("Code/hour.csv")
```

```
hour1 <- subset(hour, select = c("workingday",
                                 "temp",
                                 "hum",
                                 "windspeed",
                                 "cnt",
                                 NULL))
hour.glm <- glm(cnt ~ ., data = hour1, family = "quasipoisson")
summary(hour.glm)

##
## Call:
## glm(formula = cnt ~ ., family = "quasipoisson", data = hour1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -25.178  -10.343   -3.115    4.743   43.828
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.01970    0.03597  139.55  < 2e-16 ***
## workingday    0.03050    0.01393    2.19 0.028568 *
## temp          1.82930    0.03359   54.45  < 2e-16 ***
## hum          -1.35761    0.03528  -38.48  < 2e-16 ***
## windspeed     0.19668    0.05418    3.63 0.000284 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 134.8556)
##
##     Null deviance: 2891591  on 17378  degrees of freedom
## Residual deviance: 2158367  on 17374  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

We choose `quasipoisson` for the `family` option in `glm` function to deal with the over-dispersion problem for the bike sharing dataset. The small p values show that every covariate is significant to the model at 5% significance level. As we can see that the expected count of rented bikes in working days is greater than that in non-working days. The increase of temperature or windspeed has a positive influence to the number of rented bikes, and the increase of humidity has a negative effect on the number of rented bikes.

Next, we implement function `AdpOptSubPoi`, which is coded by Algorithm 6, to the bike sharing dataset using the following code.

```
y <- hour1$cnt
X <- cbind(1, as.matrix(hour1[, -dim(hour1)[2]]))
AdpOptSubPoi(X, y, r0 = 200, r = 500, optmethod = "A",
             delta.quant = 0.05)

##            coefficients     stdErr     Zvalue        Pvalue
## intercept   5.14630395 0.13783088  37.337816 3.997503e-305
## beta1       0.07447008 0.06144731   1.211934  2.255376e-01
## beta2       1.74470616 0.13251485  13.166118  1.374867e-39
## beta3      -1.56200797 0.15372749 -10.160889  2.963612e-24
## beta4       0.24731077 0.19582335   1.262928  2.066151e-01
```

The above result is given by setting the pilot sample size as 200 and the second stage sample size as 500 under A-optimality criterion. The option `dalta.quant` $= 0.05$ indicates that δ is chosen as the 5% quantile of $|y_i - \exp(\boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}^{\mathrm{glm},0})|$. The weighted subsample estimator is obtained by `glm` using `weights` option and the standard errors are estimated using (20).

To demonstrate the effectiveness of the proposed algorithm, we compare the MSE and running time of different methods. Table 2 shows that Algorithm 6 is better than uniform subsampling in estimation accuracy, and is computationally more efficient compared with the full data computation.

Table 2: MSE and running time of different methods for the bike sharing dataset when $n_0 = 200$ and $n_1 = 500$ are fixed for 1000 replications.

Method	MSE	CPU Seconds
Algorithm 6 optA	0.103	11.184
Algorithm 6 optL	0.116	10.727
Uniform	0.149	9.516
Full data running time: 62.379		

5 Optimal subsampling method for quantile regression

The adaptive optimal subsampling algorithm for quantile regression was discussed in Wang and Ma (2020). The quantile regression estimates a specified quantile of the response variable conditional on the covariate variable, and has form

$$q_\tau(y_i|\boldsymbol{x}_i) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta},$$

where τ represents that the τ-th quantile of y_i given $\boldsymbol{x}_i$ is measured. The full data estimator can be solved in $O(N^{5/2}d^3)$ time by interior point method (Portnoy et al., 1997). Draw a subsample with size n based on the probability distribution $\{\pi_i\}_{i=1}^N$, and record the sampled data with its subsampling probability as $\{\boldsymbol{x}_i^*, y_i^*, \pi_i^*\}_{i=1}^n$. The subsample estimator $\hat{\boldsymbol{\beta}}_{\mathrm{sub}}^{\mathrm{qr}}$ is obtained by minimizing

$$Q_{\mathrm{sub}}^*(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n \frac{(y_i^* - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^*)\{\tau - I(y_i^* - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i^* < 0)\}}{N\pi_i^*}. \tag{24}$$

The optimal subsampling probabilities under A-optimality are

$$\pi_{\mathrm{qr},i}^{\mathrm{optA}} = \frac{|\tau - I(y_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} < 0)|\|\boldsymbol{M}_Q\boldsymbol{x}_i\|}{\sum_{j=1}^N |\tau - I(y_j - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta} < 0)|\|\boldsymbol{M}_Q\boldsymbol{x}_j\|}, \quad i = 1, ..., N,$$

where $\boldsymbol{M}_Q = \frac{1}{N}\sum_{i=1}^N f_\epsilon(0, \boldsymbol{x}_i)\boldsymbol{x}\boldsymbol{x}_i^{\mathrm{T}}$ and $f_\epsilon(0, \boldsymbol{x}_i)$ is the density function of $y_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ at 0 given $\boldsymbol{x}_i$. The difficulty to estimate $f_\epsilon(0, \boldsymbol{x}_i)$ makes A-optimal subsampling probabilities hard to compute. Thus, for quantile regression, L-optimal subsampling probabilities are more favorable, which are

$$\pi_{\mathrm{qr},i}^{\mathrm{optL}}(\boldsymbol{\beta}) = \frac{|\tau - I(y_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} < 0)|\|\boldsymbol{x}_i\|}{\sum_{j=1}^N |\tau - I(y_j - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta} < 0)|\|\boldsymbol{x}_j\|}, \quad i = 1, ..., N. \tag{25}$$

The time complexity for computing $\pi_{\mathrm{qr},i}^{\mathrm{optL}}(\boldsymbol{\beta})$ is $O(Nd)$. Based on the L-optimal subsampling probabilities, the authors proposed an iteratively adaptive optimal algorithm to obtain the coefficient estimator and its estimated variance. This algorithm is stated in Algorithm 7. It has shown that the rate of convergence of the final estimator is $(n_1R)^{-1/2}$ in Wang and Ma (2020).

Algorithm 7 Iteratively adaptive optimal subsampling algorithm for quantile regression

Pilot sampling:

- Assign $\pi_i^0 = N^{-1}$ to each observation.
- Choose n_0 data points with replacement and record the subsample and associated subsampling probabilities as $\{\boldsymbol{x}_i^{*0}, y^{*0}, \pi_i^{*0}\}_{i=1}^{n_0}$.
- Obtain the pilot sample estimator $\hat{\boldsymbol{\beta}}^{\mathrm{qr},0}$ by minimizing (24) with $\{\boldsymbol{x}_i^{*0}, y^{*0}, \pi_i^{*0}\}_{i=1}^{n_0}$ plugged in.
- Calculate the approximate optimal subsampling probabilities $\hat{\pi}_{\mathrm{qr},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}^{\mathrm{qr},0})$ based on (25).

Iterative second step sampling:

for r in $\{1, 2, ..., R\}$ **do**

– Draw n_1 samples with replacement based on $\hat{\pi}_{\mathrm{qr},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}^{\mathrm{qr},0})$ and denote the subsample and corresponding subsampling probabilities as $\{\boldsymbol{x}_{r,i}^{*1}, y_{r,i}^{*1}, \pi_{r,i}^{*1}\}_{i=1}^{n_1}$.

– Obtain the subsample estimator $\hat{\boldsymbol{\beta}}_r^{\mathrm{qr}}$ by minimizing (24) with $\{\boldsymbol{x}_i^{1}, y_i^{1}, \pi_i^{*}\}_{i=1}^{n}$ replaced by $\{\boldsymbol{x}_{r,i}^{*1}, y_{r,i}^{*1}, \pi_{r,i}^{*1}\}_{i=1}^{n_1}$.

end for

Estimation: The final estimator is

$$\tilde{\boldsymbol{\beta}}^{\mathrm{qr}} = \frac{1}{R}\sum_{r=1}^{R} \hat{\boldsymbol{\beta}}_r^{\mathrm{qr}}$$

and its estimated variance-covariance matrix is

$$\tilde{\boldsymbol{V}}^{\mathrm{qr}} = \frac{1}{\nu R(R-1)}\sum_{r=1}^{R}(\tilde{\boldsymbol{\beta}}^{\mathrm{qr}} - \hat{\boldsymbol{\beta}}_r^{\mathrm{qr}})(\tilde{\boldsymbol{\beta}}^{\mathrm{qr}} - \hat{\boldsymbol{\beta}}_r^{\mathrm{qr}})^{\mathrm{T}}, \tag{26}$$

where

$$\nu = 1 - \frac{n_1 R - 1}{2}\sum_{i=1}^{N}\{\hat{\pi}_{\mathrm{qr},i}^{\mathrm{optL}}(\hat{\boldsymbol{\beta}}^{\mathrm{qr},0})\}^2.$$

5.1 Physicochemical properties of protein tertiary structure dataset

We apply the Algorithm 7 to the physicochemical properties of protein tertiary structure dataset (Dua and Graff, 2017), which contains 45730 observations and the response variable is the size of the residue ranging from 0 to 21 Armstrong. We use 8 covariates describing the features of the residue to build quantile regression model based on the dataset `casp`. The parameter estimators of `casp` are calculated with function `rq` from `quantreg` package (Koenker, 2020) by selecting option `method` = "pfn" by the following chunk of code.

```r
casp <- read.csv("Code/CASP.csv")
casp <- casp[, -which(colnames(casp) == "F3")] # F3 = F2/F1
fit.full <- rq(RMSD ~ ., tau=0.75, data = casp, method="pfn")
summary(fit.full)
```

```
##
## Call: rq(formula = RMSD ~ ., tau = 0.75, data = casp, method = "pfn")
##
## tau: [1] 0.75
##
## Coefficients:
##             Value     Std. Error t value   Pr(>|t|)
## (Intercept) 14.28730   0.44468   32.12969   0.00000
## F1           0.00135   0.00016    8.35146   0.00000
## F2           0.00363   0.00007   52.87650   0.00000
## F4          -0.14037   0.00232  -60.43679   0.00000
## F5           0.00000   0.00000   -3.92747   0.00009
## F6          -0.03302   0.00251  -13.16472   0.00000
## F7          -0.00011   0.00006   -1.79549   0.07258
## F8           0.02824   0.00094   29.91801   0.00000
## F9          -0.10077   0.00887  -11.35847   0.00000
```

From the result, we know that, at 5% significance level, the seventh covariate (Euclidian distance) is not significant to the model and all others are significant.

Algorithm 7 is realized by `QuanSub` as follows, in which the option `r0` and `r` are pilot sample size and second stage sample size, respectively, working as n_0 and n_1 in the Algorithm 7, and `RR` is the same as R in the Algorithm 7. The option `tau` $= 0.75$ indicates that we are modeling 75-th quantile of the size of the residue based on the covariates. The `optmethod` can be `L` and `uniform`, which implies optimal subsampling under L-optimality criterion and uniform subsampling, respectively.

```r
X <- cbind(1, as.matrix(casp[, -1]))
```

```
y <- casp$RMSD
QuanSub(X, y, r0 = 200, r = 1000, RR = 10, tau = 0.75,
        optmethod = "L")

##              coefficients       stdErr     Zvalue         Pvalue
## intercept  1.650621e+01 2.122373e+00   7.777244  7.412151e-15
## beta1      1.863738e-03 4.138735e-04   4.503158  6.695106e-06
## beta2      3.619006e-03 1.245316e-04  29.060951 1.119019e-185
## beta3     -1.414209e-01 6.622824e-03 -21.353567 3.612864e-101
## beta4     -7.255863e-06 2.649310e-06  -2.738774  6.166870e-03
## beta5     -3.774362e-02 8.541898e-03  -4.418645  9.932166e-06
## beta6     -4.081346e-04 1.372780e-04  -2.973052  2.948541e-03
## beta7      2.866351e-02 2.840549e-03  10.090833  6.065278e-24
## beta8     -1.329846e-01 3.951277e-02  -3.365611  7.637438e-04
```

The standard errors are obtained from (26), and z statistics and p values are to test whether the true value of corresponding parameter equals to 0 or not, where z statistics are acquired by dividing coefficient estimators by standard errors. All p values are small demonstrating that every parameter is significant under a relatively low significance level.

We also compare the performance of Algorithm 7 with uniform subsampling. Table 3 indicates that, comparing with the uniform subsampling, Algorithm 7 is more efficient in estimation accuracy. Even though Algorithm 7 takes more time in computing than uniform subsampling, it is still computationally more efficient compared with full data calculation.

Table 3: MSE and running time of different methods for physicochemical properties of protein tertiary structure dataset when $n_0 = 200$ and $n_1 = 1000$ are fixed for 1000 replications.

Method	MSE	CPU Seconds
Algorithm 7	3.464	62.113
Uniform	4.718	41.921
Full data running time: 121.077		

6 Summary

In this paper, we demonstrate the effectiveness of the optimal subsampling methods to reduce the computational burden for massive datasets, and illustrate the application of the optimal subsampling methods to logistic regression, generalized linear models and quantile regression by real data examples. The coefficient estimators obtained by the optimal subsampling methods always maintain nice statistical properties, such as consistency and asymptotic normality, making it possible to perform statistical inferences, including making hypothesis tests and constructing confidence intervals, based on the subsample.

This review focuses on the application of optimal subsampling methods, and the discussion mainly focuses on presenting optimal subsampling probabilities and practical algorithms. Theoretical properties of the resultant coefficient estimators are not discussed in details. In practical applications, problems more complex than what we have discussed can occur, and further efforts are necessary to develop suitable sampling approaches. Subsampling for big data is a promising method for estimation efficiency and computational efficiency tradeoffs. It is quite now, and much work is needed. We hope this review can be a starting point for practitioners to use the optimal subsampling methods.

Supplementary Information

The $\mathcal{R}$ functions mentioned in the paper for the optimal subsampling algorithms and all datasets can be found on the *Journal of Data Science* website.

References

Ai, M., J. Yu, H. Zhang, and H. Wang (2019). Optimal subsampling algorithms for big data regressions. *Statistica Sinica, DOI: 10.5705/ss.202018.0439*.

Cheng, Q., H. Wang, and M. Yang (2020). Information-based optimal subdata selection for big data logistic regression. *Journal of Statistical Planning and Inference*.

Derezinski, M., M. K. Warmuth, and D. J. Hsu (2018). Leveraged volume sampling for linear regression. In *Advances in Neural Information Processing Systems*, pp. 2505–2514.

Drineas, P., M. Mahoney, S. Muthukrishnan, and T. Sarlos (2011). Faster least squares approximation. *Numerische Mathematik 117*, 219–249.

Drineas, P., M. W. Mahoney, and S. Muthukrishnan (2006). Sampling algorithms for l_2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1127–1136. Society for Industrial and Applied Mathematics.

Dua, D. and C. Graff (2017). UCI machine learning repository.

Fanaee-T, H. and J. Gama (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 1–15.

Fithian, W. and T. Hastie (2014). Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics 42*(5), 1693.

Han, L., K. M. Tan, T. Yang, and T. Zhang (2020). Local Uncertainty Sampling for Large-Scale Multi-Class Logistic Regression. *The Annals of Statistics*, in press.

Koenker, R. (2020). *quantreg: Quantile Regression.* R package version 5.55.

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

Lumley, T. (2020). survey: analysis of complex survey samples. R package version 4.0.

Ma, P., M. W. Mahoney, and B. Yu (2015). A statistical perspective on algorithmic leveraging. *The Journal of Machine Learning Research 16*(1), 861–911.

Ma, P. and X. Sun (2015). Leveraging for big data regression. *Wiley Interdisciplinary Reviews: Computational Statistics 7*(1), 70–76.

Ma, P., X. Zhang, X. Xing, J. Ma, and M. Mahoney (2020, 26–28 Aug). Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. In S. Chiappa and R. Calandra (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Volume 108 of *Proceedings of Machine Learning Research*, Online, pp. 1026–1035. PMLR.

Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning 3*(2), 123–224.

Portnoy, S., R. Koenker, et al. (1997). The gaussian hare and the laplacian tortoise: computability of squared-error versus absolute-error estimators. *Statistical Science 12*(4), 279–300.

Pronzato, L. and H. Wang (2020). Sequential online subsampling for thinning experimental designs. *arXiv preprint arXiv:2004.00792*.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Schifano, E. D., J. Wu, C. Wang, J. Yan, and M.-H. Chen (2016). Online updating of statistical inference in the big data setting. *Technometrics 58*(3), 393–403.

Toulis, P., E. M. Airoldi, et al. (2017). Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics 45*(4), 1694–1727.

Wang, H. (2019a). Divide-and-conquer information-based optimalsubdata selection algorithm. *Journal of Statistical Theory and Practice 13*(3), 46.

Wang, H. (2019b). More efficient estimation for logistic regression with optimal subsamples. *Journal of Machine Learning Research 20*(132), 1–59.

Wang, H. and Y. Ma (2020). Optimal subsampling for quantile regression in big data. *Biometrika*, in press.

Wang, H., M. Yang, and J. Stufken (2019). Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association 114*(525), 393–405.

Wang, H., R. Zhu, and P. Ma (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association 113*(522), 829–844.

Yao, Y. and H. Wang (2018). Optimal subsampling for softmax regression. *Statistical Papers*, 585–599.

Yu, J., H. Wang, M. Ai, and H. Zhang (2020). Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association* (just-accepted), 1–29.