



Article

Response-Aided Score-Matching Representative Approaches for Big Data Analysis and Model Selection under Generalized Linear Models [†]

Duo Zheng ¹, Keren Li ²  and Jie Yang ^{3,*} ¹ Amazon, 425 106th Ave NE, Bellevue, WA 98004, USA; zhengduo2010@gmail.com² Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL 35294, USA; kli@uab.edu³ Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA

* Correspondence: jyang06@uic.edu

[†] This work was performed while at the University of Illinois at Chicago and is not associated with Amazon.

Abstract: In this paper, we propose an efficient method called the response-aided score-matching representative (RASMR) approach to facilitate massive data model selection and data analysis with generalized linear models (GLMs) and a predetermined data partition due to data localization. Similar to the original score-matching representative (SMR) approach, RASMR constructs an artificial data point, called the representative, for each data block. It then fits a GLM on the representative dataset, which provides not only an efficient approach for massive data analysis but also an ideal solution in response to privacy concerns by avoiding the transfer of sensitive data. By further splitting the data blocks according to the values of the response variables, RASMR can obtain more accurate parameter estimates than SMR. Furthermore, by theoretical justifications and simulation studies, we show that RASMR can be more efficiently utilized for model selection and variable selection for a massive dataset by approximating the Akaike information criterion (AIC) and the aggregated prediction errors for cross-validation, which are commonly used for choosing the most appropriate statistical model and drawing reliable conclusions. We also apply the proposed RASMR approach to the airline on-time performance data, which consists of 371 data files labeled by month, and show that RASMR can be successfully used for selecting the most appropriate model for real massive data analysis.

Keywords: cross-validation; data localization; distributed data; model selection; variable selection



Citation: Zheng, D.; Li, K.; Yang, J. Response-Aided Score-Matching Representative Approaches for Big Data Analysis and Model Selection under Generalized Linear Models.

Algorithms **2024**, *17*, 456. <https://doi.org/10.3390/a17100456>

Academic Editor: Takeshi Yamada

Received: 5 August 2024

Revised: 22 September 2024

Accepted: 9 October 2024

Published: 14 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The numerous innovations in data analysis in the last decade have dramatically affected the technologies used in our daily lives. Datasets with unprecedented sizes and complexities are rapidly generated and collected from a great variety of resources [1]. While big datasets bring us incredible opportunities for new discoveries, many traditional methods that perform well for moderate sample sizes are no longer realistic for analyzing massive amounts of data [2].

To address the big data challenges, many traditional statistical tools have been reinvented or adapted to deal with the gigantic volume or size of big data, which is a major goal of our work. Comprehensive reviews, such as [3,4], have been provided for algorithmic solutions to big data problems, including divide-and-conquer, subsampling-based approaches, stochastic gradient descent, and online updating.

Divide-and-conquer approaches split the big data into manageable blocks, extract local summaries from each data block, and then generate overall insight. Various types of algorithms have been proposed to adapt classic statistical tools to big data problems [5–12].

Subsampling-based approaches draw subsamples from the original data by elaborately designed sampling mechanisms and then perform downstream inference and prediction

based on the subsamples. For example, leveraging for big data regression, [13] constructed nonuniform sampling probabilities so that influential data points were sampled with high probabilities, and [14] proposed a novel method called information-based optimal sub-data selection (IBOSS), which selects samples from a big dataset based on the D-optimality criterion; it has been extended to various statistical models, such as logistic regression [15].

When multiple subsamples of the original data are available, different approaches have been proposed to aggregate the estimates obtained from different subsamples, including bagging or bootstrap aggregating [16], stacking [17,18], maging or maximin aggregating [19], neagging or normalized entropy aggregating [20,21]. These aggregation approaches are especially useful for inhomogeneous large-scale data under regression analysis.

Stochastic gradient descent (SGD) algorithms [22–24] update in a sequential manner based on a noisy gradient. Local SGD is the key building block of different federated learning algorithms [25,26], which were discussed for both homogeneous nodes [27–32] and heterogeneous nodes [29,33–42].

As stated in [43], new challenges in the big data era involve data security and localization legislation. According to [44,45], legislation on data localization has emerged as a global trend, with more nations developing laws that prohibit the transfer of sensitive data. Data localization may not kill global cooperation in data science, but it will undoubtedly create unpleasant barriers to data communication. A novel approach called the score-matching representative (SMR) was proposed by [43] for analyzing big data under communication restraints. Given an existing data partition, such as data blocks labeled by countries, regions, or sources, SMR constructs model-specified data representative(s) for each block and performs downstream analysis on the constructed representatives. Unlike the divide-and-conquer or subsampling approaches, SMR does not require communicating the actual data but rather their representatives, which are not part of the original data and, thus, avoid their transfer. Comprehensive studies show that the accuracy of the estimated model parameters based on SMR can be comparable to the full data estimates [43]. According to [43], the computational complexity of SMR is $O(Np)$ given the sample size N and the number p of model parameters.

Nevertheless, there are three hidden traps in [43]’s SMR method (see Section 2.1 for more details), as follows: (1) there can be multiple solutions to the score-matching equations; (2) the constructed representatives may be quite far away from their data blocks; and (3) the SMR algorithm may fail to converge. Due to these issues, SMR may not be accurate enough for more elaborate tasks, such as model selection and variable selection.

In this paper, we develop a new representative approach, called the *response-aided score-matching representative* (RASMR) approach, for big data analysis under generalized linear models (GLMs). It improves SMR significantly by splitting the data blocks further with the aid of the response variable (see Figure 1 for a graphical display). Since the refined data blocks yield only one solution to the score-matching equation, RASMR ensures the uniqueness of the data representatives. Compared with SMR, RASMR can not only provide much more accurate estimates for model parameters but also be used for model selection and variable selection efficiently, which are critical for data scientists to draw reliable conclusions from data analysis.

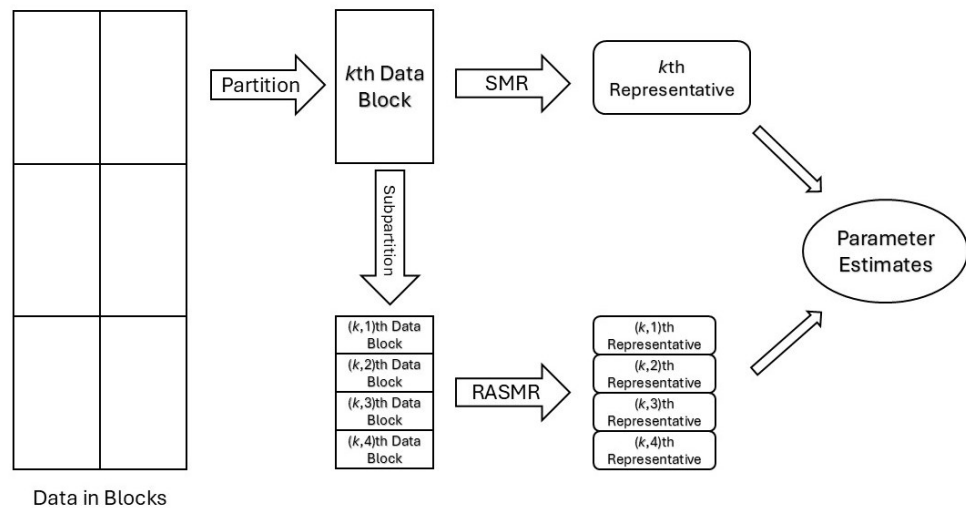


Figure 1. Graphical display for SMR and RASMR.

2. RASMR for Big Data Analysis under GLM

2.1. SMR Approach for GLM

In this section, we review the key components of the original SMR approach proposed by [43], along with its potential issues.

Following the notation of [43], the original data $\{(x_i, y_i), i = 1, \dots, N\}$ are provided with an index partition $\{I_1, \dots, I_K\}$ of $I = \{1, \dots, N\}$, where $x_i \in \mathbb{R}^d$ represents the i th covariate vector, $y_i \in \mathbb{R}$ is the corresponding response, and $I = \cup_{j=1}^K I_j$, $I_j \neq \emptyset$ for each j , and $I_i \cap I_j = \emptyset$ for each $i \neq j$.

Under a generalized linear model (GLM) [46,47], there is a link function g , p known predictor functions h_1, \dots, h_p , and p unknown regression parameters β_1, \dots, β_p , such that the expectation of the response variable Y_i given x_i satisfies the following:

$$E(Y_i | x_i) = \mu_i \text{ and } \eta_i = g(\mu_i) = \mathbf{X}_i^T \boldsymbol{\beta}, \tag{1}$$

where $\mathbf{X}_i = (h_1(x_i), \dots, h_p(x_i))^T$, $i = 1, \dots, N$, and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T$.

Given the GLM (1), we denote the k th data block by $\mathcal{D}_k = \{(\mathbf{X}_i, y_i), i \in I_k\}$ and let $s_k(\boldsymbol{\beta}) = \sum_{i \in I_k} [y_i - G(\eta_i)]v(\eta_i)\mathbf{X}_i$ be the contribution made by the k th data block to the score function $s(\boldsymbol{\beta}) = \sum_{k=1}^K s_k(\boldsymbol{\beta})$ [43,46], where $G(\eta) = g^{-1}(\eta)$, $v(\eta) = G'(\eta)/h(\eta)$, and $h(\eta_i) = \text{Var}(Y_i)$ are functions of η or η_i . According to Section 2.5 of [46], the maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ solves the score equation $s(\boldsymbol{\beta}) = 0$. The SMR algorithm [43] was designed to find $\tilde{y}_k \in \mathbb{R}$ and $\tilde{\mathbf{X}}_k \in \mathbb{R}^p$ solving $s_k(\boldsymbol{\beta}) = n_k[\tilde{y}_k - G(\tilde{\eta}_k)]v(\tilde{\eta}_k)\tilde{\mathbf{X}}_k \triangleq \tilde{s}_k(\boldsymbol{\beta})$, where $n_k = |I_k|$ is the size of I_k , and $\tilde{\eta}_k = \tilde{\mathbf{X}}_k^T \boldsymbol{\beta}$. More specifically, the SMR algorithm first chooses the following:

$$\tilde{y}_k = \left[\sum_{i \in I_k} v(\eta_i)\eta_i \right]^{-1} \sum_{i \in I_k} v(\eta_i)\eta_i y_i, \tag{2}$$

then solves the score-matching equation as follows:

$$n_k v(\tilde{\eta}_k)[\tilde{y}_k - G(\tilde{\eta}_k)]\tilde{\eta}_k = \sum_{i \in I_k} v(\eta_i)[y_i - G(\eta_i)]\eta_i \tag{3}$$

for $\tilde{\eta}_k \in \mathbb{R}$, and then constructs the k th representative $(\tilde{\mathbf{X}}_k, \tilde{y}_k)$ by calculating the following:

$$\tilde{\mathbf{X}}_k = \{n_k v(\tilde{\eta}_k)[\tilde{y}_k - G(\tilde{\eta}_k)]\}^{-1} \sum_{i \in I_k} v(\eta_i)[y_i - G(\eta_i)]\mathbf{X}_i. \tag{4}$$

Then the score function $s(\beta) = \sum_{i=1}^N [y_i - G(\eta_i)]v(\eta_i)\mathbf{X}_i = \sum_{k=1}^K s_k(\beta) = \sum_{k=1}^K \tilde{s}_k(\beta) \triangleq \tilde{s}(\beta)$. The MLE $\hat{\beta}$ of β based on the full data, which solves the score equation $s(\beta) = 0$, is expected to be the same as the SMR estimate $\tilde{\beta}$ based on the weighted representative data $\{(n_k, \tilde{\mathbf{X}}_k, \tilde{y}_k), k = 1, \dots, K\}$, which solves the matched score equation $\tilde{s}(\beta) = 0$. In practice, since β is unknown, the SMR algorithm first solves the matched score function $\tilde{s}(\beta) = 0$ with some initial representatives for $\tilde{\beta}$, and then solves the score-matching Equation (3) for the representatives. The procedure may continue iteratively until reaching a predetermined accuracy level. Three iterations were suggested by [43] for general applications, whose computational complexity is $O(Np)$.

Given the successful applications of SMR in estimating parameter values for big data analysis using a GLM, its accuracy level may not be high enough for more delicate applications, including model selection and variable selection, due to the following three issues. Firstly, there can be more than one solution for Equation (3) (see Section 2.2). SMR chooses the solution whose representative is closest to the mean representative (MR, that is, $\bar{\mathbf{X}}_k = n_k^{-1} \sum_{i \in I_k} \mathbf{X}_i$, $\bar{y}_k = n_k^{-1} \sum_{i \in I_k} y_i$), which, however, may not match the likelihood well. Secondly, the predictor representative $\tilde{\mathbf{X}}_k$ obtained from Equation (4) may be far away from its corresponding data block (see Section S2 in the Supplementary Materials), which may not represent its data block well. Thirdly, the SMR algorithm may not converge well in practice, which may occur along with misspecified link functions or highly skewed predictor distributions (see Section 4).

2.2. Solving the Score-Matching Equation with Splitting Points

In this section, we investigate the number of solutions for the score-matching Equation (3) and explain why more stable and accurate solutions can be obtained for Equation (3) by further splitting the data blocks, which is our motivation to propose the RASMR algorithms in later sections.

According to Equations (2) and (4), the k th representative $(\tilde{\mathbf{X}}_k, \tilde{y}_k)$ can be obtained as weighted averages of the k th data block $\mathcal{D}_k = \{(\mathbf{X}_i, y_i), i \in I_k\}$. The weights of y_i 's and \mathbf{X}_i 's are $v(\eta_i)\eta_i$ and $v(\eta_i)[y_i - G(\eta_i)]$, respectively. To stabilize the constructed representatives, we propose to keep the weights with the same sign (that is, all positive or all negative) in each individual data block. That is, besides splitting the data block according to $\text{sgn}(\eta_i)$, as suggested in Remark 3.1 of [43], we suggest further splitting the data blocks by $\text{sgn}(y_i - G(\eta_i))$.

Now, we investigate the number of solutions to Equation (3) and how to further split the data block to ensure a unique solution. Following [43], we denote $S(\eta) = v(\eta)[\tilde{y}_k - G(\eta)]\eta$. According to the proof for Theorem 3.1 in [43], Equation (3) can be rewritten as follows:

$$S(\tilde{\eta}_k) = \frac{1}{n_k} \sum_{i \in I_k} S(\eta_i). \tag{5}$$

We further denote $\eta_k^\wedge = \min_{i \in I_k} \{\eta_i\}$ and $\eta_k^\vee = \max_{i \in I_k} \{\eta_i\}$. If both $v(\eta)$ and $G(\eta)$ are continuous, then there exists a solution $\eta_* \in [\eta_k^\wedge, \eta_k^\vee]$ that solves Equation (5) (see Theorem 3.1 in [43]).

Examples of $v(\eta)$ and $G(\eta)$ for commonly used GLMs can be found in Table 1 of [43], which are all continuous. If $S(\eta)$ is strictly monotone on $[\eta_k^\wedge, \eta_k^\vee]$, then there exists a unique $\eta_* \in [\eta_k^\wedge, \eta_k^\vee]$ that solves (3) (see Theorem A1 in Appendix A).

If $v(\eta)$ is a constant, such as for the normal model with an identity link, the Bernoulli model with a logit link, the Poisson model with a log link, the gamma model with a reciprocal link, and the inverse Gaussian model with an inverse-square link, without any loss of generality, we rewrite $S(\eta) = [\tilde{y}_k - G(\eta)]\eta$. Then, its first derivative $S'(\eta) = \tilde{y}_k - [G(\eta) + G'(\eta)\eta]$, with its key component $T(\eta) \triangleq G(\eta) + G'(\eta)\eta$. In this case, we have the following: $\tilde{y}_k = \frac{\sum_{i \in I_k} \eta_i y_i}{\sum_{i \in I_k} \eta_i} = \frac{n_k^{-1} \sum_{i \in I_k} \eta_i y_i}{\bar{\eta}_k}$, where $\bar{\eta}_k = n_k^{-1} \sum_{i \in I_k} \eta_i$ corresponds to the

mean representative (MR). We denote the following: $\tilde{G}_k = \frac{\sum_{i \in I_k} \eta_i G(\eta_i)}{\sum_{i \in I_k} \eta_i} = \frac{n_k^{-1} \sum_{i \in I_k} \eta_i G(\eta_i)}{\bar{\eta}_k}$. Then, we have the following:

$$\bar{S} \triangleq \frac{1}{n_k} \sum_{i \in I_k} S(\eta_i) = \frac{1}{n_k} \sum_{i \in I_k} [\tilde{y}_k - G(\eta_i)] \eta_i = \bar{\eta}_k (\tilde{y}_k - \tilde{G}_k). \tag{6}$$

For the normal model with an identity link, that is, the usual linear model, there can be up to two solutions to (3) (see Theorem A2 in Appendix A).

For Bernoulli models (see Table 1 of [43]), $y_i \in \{0, 1\}$ and $G(\eta_i) \in (0, 1)$ for all i . Then, $y_i < G(\eta_i)$ always implies $y_i = 0$, and $y_i > G(\eta_i)$ always implies $y_i = 1$. Suppose (i) either $\eta_i > 0$ for all $i \in I_k$ or $\eta_i < 0$ for all $i \in I_k$; and (ii) either $y_i > G(\eta_i)$ for all $i \in I_k$ or $y_i < G(\eta_i)$ for all $i \in I_k$. Then, \tilde{y}_k is either 0 or 1. Depending on $\tilde{y}_k = 0$ or 1, we denote $S(\eta)$ as $S_0(\eta)$ or $S_1(\eta)$, and the potential splitting point as η_l or η_r (see Table 1), respectively. For Bernoulli models with logit, probit, cloglog, loglog, or cauchit links, a data block I_k after splitting at η_l or η_r yields a unique solution solving (3) (see Theorem A3 in Appendix A and Lemmas S1–S5 in Section S6 of the Supplementary Materials). We summarize in Table 1 the potential splitting points η_l for blocks with $\tilde{y}_k = 0$ and η_r for blocks with $\tilde{y}_k = 1$.

Table 1. Splitting points for Bernoulli models with different links.

Link Function	η_l for $S_0(\eta)$	η_r for $S_1(\eta)$
logit	−1.278464542761	1.278464542761
probit	−0.839923675692	0.839923675692
cloglog	−1	0.729114174900
loglog	−0.729114174900	1
cauchit	−0.801916425045	0.801916425045

For the Poisson model with a log link, if either $\eta_i > 0$ for all $i \in I_k$ or $\eta_i < 0$ for all $i \in I_k$, then there are up to two solutions solving (3) (see Theorem A4 in Appendix A).

For the gamma model with a reciprocal link, $\bar{\eta}_k = n_k^{-1} \sum_{i \in I_k} \eta_i$ is the unique solution solving (3) (see Theorem A5 in Appendix A).

For the inverse Gaussian model with an inverse-square link, in general, there are up to two solutions solving (3) (see Theorem A6 in Appendix A).

In conclusion, the score-matching Equation (3) often yields two solutions for commonly used GLMs, confirming the first hidden trap mentioned in the Introduction section for the original SMR algorithm. On the other hand, there are, at most, two solutions for Equation (3) (see Theorems A2–A6 in Appendix A), which motivates us to further split the corresponding data block in a way that each sub-block yields a unique solution.

2.3. Response-Aided Score-Matching Representative Approach

In this section, we propose a new algorithm (see Algorithm 1 for its pseudocode) with further splits based on the response variable y_i 's, and call it the response-aided score-matching representative (RASMR) approach. This suggests that further splits may at most quadruple the original number of data blocks. Since the time complexity of the original SMR is $O(Np)$ [43], which does not depend on the number of data blocks, the RASMR algorithm consumes no significantly more time than the SMR algorithm (see Section 4.2). Its time complexity is $O(Np)$ as well.

Algorithm 1: RASMR.

Data: $\mathcal{D} = \{(\mathbf{X}_i, y_i), i = 1, \dots, N\}$ with a partition of K blocks indexed by $\{I_1, \dots, I_K\}$. Denote the k th data block $\mathcal{D}_k = \{(\mathbf{X}_i, y_i), i \in I_k\}$.

Result: Parameter estimate $\tilde{\beta}$ of a given GLM with a predetermined number T of iterations.

Calculate the initial weighted representative data: $\tilde{\mathcal{D}}^{(0)} = \{(n_k, \tilde{\mathbf{X}}_k^{(0)} = \bar{\mathbf{X}}_k, \tilde{y}_k^{(0)} = \bar{y}_k)\}_{k=1}^K$ with $\bar{\mathbf{X}}_k = n_k^{-1} \sum_{i \in I_k} \mathbf{X}_i$ and $\bar{y}_k = n_k^{-1} \sum_{i \in I_k} y_i$, that is, the mean representatives;

Implement the iteratively reweighted least squares (IRLS) procedure [48] on $\tilde{\mathcal{D}}^{(0)}$ to obtain the initial estimate $\tilde{\beta}^{(0)}$;

for $t = 1, \dots, T$ **do**

for $k = 1, \dots, K$ **do**

(1) Compute $\eta_i = \mathbf{X}_i^T \tilde{\beta}^{(t-1)}$ for each $i \in I_k$;

(2) Split \mathcal{D}_k to sub-blocks by (2.1) the sign of η_i ; (2.2) further by the sign of $y_i - G(\eta_i)$;

(3) Suppose \mathcal{D}_k is split to $\{\mathcal{D}_{kl}, l = 1, \dots, m_k\}$ with index blocks $I_{kl} \subseteq I_k$.

for each I_{kl} **do**

(3.1) compute $\tilde{y}_{kl}^{(t)}$ by (2);

(3.2) solve (3) for $\tilde{\eta}_{kl}^{(t)}$;

(3.3) **while** $\tilde{\eta}_{kl}^{(t)}$ is not unique **do**

compute $\tilde{\eta}_{kl-MAX}^{(t)} = \arg \max_{\tilde{\eta}_{kl}} v(\tilde{\eta}_{kl})[\tilde{y}_{kl}^{(t)} - G(\tilde{\eta}_{kl})]\tilde{\eta}_{kl}$;

split \mathcal{D}_{kl} further by the sign of $\eta_i - \tilde{\eta}_{kl-MAX}^{(t)}$, and then return to (3.1);

end

(3.4) compute $\tilde{\mathbf{X}}_{kl}^{(t)}$ by (4);

end

end

Implement the IRLS on the updated weighted representative dataset:

$\tilde{\mathcal{D}}^{(t)} = \{(n_{kl}, \tilde{\mathbf{X}}_{kl}^{(t)}, \tilde{y}_{kl}^{(t)})\}, k = 1, \dots, K; l = 1, \dots, m_k\}$ to obtain $\tilde{\beta}^{(t)}$;

end

Report $\tilde{\beta} = \tilde{\beta}^{(T)}$

Remark 1. The iteratively reweighted least squares (IRLS or IWLS) procedure has been widely used for finding the MLE of a GLM [48–50]. Nevertheless, some more robust variants of IRLS have been proposed to make the estimates less sensitive to outliers (see [51] and references therein). One may choose a more robust procedure than IRLS in Algorithm 1, given that the targeted parameter estimate $\tilde{\beta}$ is under the same criterion.

Remark 2. Following the splits described in (2) of Algorithm 1, according to Theorems A2–A6 in Appendix A, many sub-blocks yielded a unique solution to (3). There are still leftover cases for normal/linear, Bernoulli, Poisson, and inverse Gaussian models, under which, some sub-blocks may yield up to two solutions to (3). For those cases, according to the proofs of Theorems A2–A4 and A6 (see Section S6 in the Supplementary Materials), we may further split such a block into two sub-blocks according to the peak value η of $S(\eta)$, which is $\frac{1}{2}\tilde{y}_{kl}$ for normal/linear models, η_l or η_r for Bernoulli models, $u(\tilde{y}_{kl})$ for the Poisson models, or $(4\tilde{y}_{kl}^2)^{-1}$ for the inverse Gaussian models. Based on our experience, such a split often leads to data blocks with a unique solution to (3).

2.4. RASMR with the Delta Ratio Split

As we will demonstrate in Section 4.2, RASMR significantly improves upon SMR in estimating model parameters. However, it may not perform sufficiently well when approximating likelihood with non-Gaussian covariates \mathbf{x}_i or predictors \mathbf{X}_i for model selection purposes.

To investigate this, we start with the mean representative (MR) of the predictors in block I_k , $\bar{\mathbf{X}}_k = n_k^{-1} \sum_{i \in I_k} \mathbf{X}_i$, which is a natural choice for the block center. Then, the predictor radius of I_k can be defined as $\Delta_k = \max_{i \in I_k} \|\mathbf{X}_i - \bar{\mathbf{X}}_k\|$, where $\|\cdot\|$ is the Euclidean norm.

We call the relative distance of $\tilde{\mathbf{X}}_k$ from $\bar{\mathbf{X}}_k$, the delta ratio, defined as $\tilde{\delta}_k = \|\tilde{\mathbf{X}}_k - \bar{\mathbf{X}}_k\| / \Delta_k$.

By definition, $\tilde{\delta}_k = 0$ if $n_k = 1$. If $\tilde{\delta}_k > 1$ for some I_k , then its predictor representative $\tilde{\mathbf{X}}_k$ is outside the corresponding data block, which implies that it may not be a good representative for calculating the likelihood. Compared with SMR, RASMR’s delta ratios

are significantly smaller (see Figure S1 in the Supplementary Materials), which partly explains why RASMR improves SMR significantly. Nevertheless, the delta ratios of RASMR may tend to inflate when the distribution of covariates is too extreme or complicated (see Section S2 in the Supplementary Materials for more details).

If $\tilde{\delta}_k$ is greater than a predefined threshold δ_0 , a fourth layer of further splitting at the mean $\bar{\eta}_k = n_k^{-1} \sum_{i \in I_k} \eta_i$ of η_i 's may be applied. Since the sample mean is sensitive to outliers, using $\bar{\eta}_k$ as the cutoff point of the split may enable us to separate the outliers from the majority of the data block, which will make the leftover members in the data blocks closer to each other. In other words, we may identify the minorities of data points in the original data block and separate them into their own data block.

In practice, if the likelihood approximation is not a serious concern, setting the threshold for delta ratios to be one is a conservative choice, since it only requires the representatives to stay inside its data block, not necessarily very close to the center (see Section 4.3 for further discussion on choosing δ_0). The pseudocode of the RASMR algorithm with the delta ratio split is described in Algorithm 2.

Algorithm 2: RASMR algorithm with the delta ratio split.

Data: $\mathcal{D} = \{(\mathbf{X}_i, y_i), i = 1, \dots, N\}$ with a partition of K blocks indexed by $\{I_1, \dots, I_K\}$. Denote $\mathcal{D}_k = \{(\mathbf{X}_i, y_i), i \in I_k\}$ as the k th data block. Set the delta ratio threshold $\delta_0 > 0$ (e.g., $\delta_0 = 1$).

Result: Parameter estimate $\tilde{\beta}$ of a given GLM with a predetermined number T of iterations. Generate the mean representative for each data block to form the initial weighted data:

$$\tilde{\mathcal{D}}^{(0)} = \{(n_k, \tilde{\mathbf{X}}_k^{(0)} = \bar{\mathbf{X}}_k, \tilde{y}_k^{(0)} = \bar{y}_k)\}_{k=1}^K;$$

Implement the IRLS on $\tilde{\mathcal{D}}^{(0)}$ for the initial estimate $\tilde{\beta}^{(0)}$;

for $t = 1, \dots, T$ **do**

for $k = 1, \dots, K$ **do**

(1) Compute $\eta_i = \mathbf{X}_i^T \tilde{\beta}^{(t-1)}$ for each $i \in I_k$;

(2) Split \mathcal{D}_k by the sign of η_i and the sign of $y_i - G(\eta_i)$;

(3) Suppose \mathcal{D}_k is split to $\{\mathcal{D}_{kl}, l = 1, \dots, m_k\}$ with index blocks $I_{kl} \subseteq I_k$.

for each I_{kl} **do**

(3.1) compute $\tilde{y}_{kl}^{(t)}$ by (2);

(3.2) solve (3) for $\tilde{\eta}_{kl}^{(t)}$;

(3.3) **while** $\tilde{\eta}_{kl}^{(t)}$ is not unique **do**

compute $\tilde{\eta}_{kl-MAX}^{(t)} = \arg \max_{\tilde{\eta}_{kl}} v(\tilde{\eta}_{kl}) [\tilde{y}_{kl}^{(t)} - G(\tilde{\eta}_{kl})] \tilde{\eta}_{kl}$;

split \mathcal{D}_{kl} further by the sign of $\eta_i - \tilde{\eta}_{kl-MAX}^{(t)}$, and then return to (3.1);

end

(3.4) compute $\tilde{\mathbf{X}}_{kl}^{(t)}$ by (4);

(3.5) compute $\tilde{\mathbf{X}}_{kl}, \Delta_{kl} = \max_{i \in I_{kl}} \|\mathbf{X}_i - \tilde{\mathbf{X}}_{kl}\|$, and the delta ratio $\tilde{\delta}_{kl} = \|\tilde{\mathbf{X}}_{kl}^{(t)} - \tilde{\mathbf{X}}_{kl}\| / \Delta_{kl}$;

(3.6) **while** $\tilde{\delta}_{kl} > \delta_0$ **do**

split \mathcal{D}_{kl} by the sign of $\eta_i - \tilde{\eta}_{kl}$, where $\tilde{\eta}_{kl} = n_{kl}^{-1} \sum_{i \in I_{kl}} \eta_i$, and then return to (3.1);

end

end

end

Implement the IRLS on the weighted dataset $\tilde{\mathcal{D}}^{(t)} = \{(n_{kl}, \tilde{\mathbf{X}}_{kl}^{(t)}, \tilde{y}_{kl}^{(t)})\}_{k=1, \dots, K; l=1, \dots, m_k}$ to obtain $\tilde{\beta}^{(t)}$;

end

Report $\tilde{\beta} = \tilde{\beta}^{(T)}$

2.5. Learning Rate Scheduling

As an iterative method to maximize the log-likelihood function, the RASMR approach may suffer from a convergence issue, since the solution at each iteration may be far away from the global maximizer, or jump back and forth at the final stages (see Figure S13 in the Supplementary Materials for such an example).

The convergence issue of RASMR comes from various reasons. It could be caused by a bad initial estimate via the mean representative (MR). A small number of data blocks may also make the convergence poor (see Figure S12 in the Supplementary Materials for such an example; see also Section S1 for generating data blocks), especially when the number of

parameters is moderately large (see Figure S13). The inherent complexities of the original data, such as ultra-high-dimensional, highly skewed, or heavily tailed distributions of covariates or predictors, may cause similar problems as well.

To make the RASMR approach more robust and practically useful for big data analysis, we adopt a learning rate scheduling strategy, which is a popular idea in the modern development of machine learning and artificial intelligence to resolve convergence issues (see, e.g., [52,53], and references therein). More specifically, we update the RASMR parameter estimates with an exponential learning rate decay as follows:

$$\beta^{(t)} = \beta^{(t-1)} + e^{-\theta t} (\tilde{\beta}^{(t)} - \beta^{(t-1)}), \tag{7}$$

where $\tilde{\beta}^{(t)}$ is the original RASMR parameter estimate after the t th iteration, $\theta > 0$ is a hyperparameter to control the learning rate $e^{-\theta t}$. Apparently, if $\theta = 0$ or with the learning rate 1, (7) yields the original RASMR estimates. A larger value of θ posts a quicker rate of decay, and will force the convergence of iterative parameter estimates. On the other hand, if the learning rate quickly shrinks to zero, the estimate may be updated very slowly, making the improvement of later iterations negligible. To overcome the issue that the learning rate decays too fast, we suggest truncating the learning rate at, say, the 10th iteration and making the learning rate constant afterward. The truncation strategy for the exponential learning rate decay works well when the number of variables is as large as 100 (see Figure S13).

3. Model Selection and Variable Selection Using RASMR

3.1. Information-Based Criteria and Model Selection

During the process of RASMR parameter estimation by Algorithms 1 and 2, two very useful side products are also generated, namely the response representative \tilde{y}_k and the predictor representative $\tilde{\mathbf{X}}_k$. The quadruple $(n_k, \tilde{y}_k, \tilde{\mathbf{X}}_k, \tilde{\beta})$ provides information for approximating the maximum likelihood, which is a key component for information-based criteria, such as the Akaike information criterion (AIC, [54,55]) and Bayesian information criterion (BIC, [55]).

Recall that the log-likelihood function $l(\beta; \mathbf{y}, \mathbf{X})$ through the full dataset $\mathbf{y} = (y_1, \dots, y_N)^T$ and $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_N)^T$ attains its maximum at the MLE $\hat{\beta}$. We denote the log-likelihood approximated through the RASMR representative quadruple set $\{(n_k, \tilde{y}_k, \tilde{\mathbf{X}}_k, \tilde{\beta}), k = 1, \dots, K\}$ by $l(\tilde{\beta}; \tilde{\mathbf{y}}, \tilde{\mathbf{X}})$, where $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_K)^T$, and $\tilde{\mathbf{X}} = (\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_K)^T$.

Denoting $\tilde{\Delta} = \max_k \max_{i \in I_k} \|\mathbf{X}_i - \tilde{\mathbf{X}}_k\|$, the maximum Euclidean distance of predictor vectors away from their corresponding predictor representatives across all data blocks, the following theorem guarantees the consistency of the estimated log-likelihood $l(\tilde{\beta}; \mathbf{y}, \mathbf{X})$, as $\tilde{\Delta}$ goes to zero.

Theorem 1. *Suppose the log-likelihood function $l(\beta; \mathbf{y}, \mathbf{X})$ is twice differentiable and strictly concave on a compact set $C \subset \mathcal{R}^P$, with its maximum not located on the boundary of C . Suppose $\tilde{\beta}$ is the estimate obtained by Algorithms 1 or 2, and $\tilde{y}_k = n_k^{-1} \sum_{i \in I_k} y_i + O(\tilde{\Delta})$ for each k , then $l(\tilde{\beta}; \mathbf{y}, \mathbf{X}) - l(\hat{\beta}; \mathbf{y}, \mathbf{X}) = O(\tilde{\Delta}^{1/2})$ as $\tilde{\Delta} \rightarrow 0$.*

The proof of Theorem 1, as well as all other proofs in this paper, has been relegated to Section S6 of the Supplementary Materials.

We denote the AIC and BIC values calculated based on $l(\tilde{\beta}; \mathbf{y}, \mathbf{X})$ as \widetilde{AIC} and \widetilde{BIC} , respectively. As a direct conclusion of Theorem 1, we have the following corollary:

Corollary 1. *For a generalized linear model under the same technical conditions of Theorem 1, the values of \widetilde{AIC} and \widetilde{BIC} converge to the original values of AIC and BIC, respectively, as $\tilde{\Delta} \rightarrow 0$. Moreover, $\widetilde{AIC} = AIC + O(\tilde{\Delta}^{1/2})$, and $\widetilde{BIC} = BIC + O(\tilde{\Delta}^{1/2})$.*

The computation of $l(\tilde{\beta}; \tilde{y}, \tilde{X})$ only requires the K representatives rather than the whole dataset. Its accuracy not only depends on the estimate $\tilde{\beta}$ of the parameters but also on the representatives \tilde{y} and \tilde{X} . The next theorem confirms that the estimated log-likelihood $l(\tilde{\beta}; \tilde{y}, \tilde{X})$ based on RASMR algorithms is consistent as $\tilde{\Delta}$ goes to zero.

Theorem 2. *Under the same technical conditions of Theorem 1, $l(\tilde{\beta}; \tilde{y}, \tilde{X}) - l(\hat{\beta}; y, X) = O(\tilde{\Delta}^{1/2})$, as $\tilde{\Delta} \rightarrow 0$.*

We denote the AIC and BIC calculated with $l(\tilde{\beta}; \tilde{y}, \tilde{X})$ as \widetilde{AIC} and \widetilde{BIC} , respectively. As a direct conclusion of Theorem 2, we have the following corollary:

Corollary 2. *For a generalized linear model under the same technical conditions of Theorem 1, the values of \widetilde{AIC} and \widetilde{BIC} converge to the original values of AIC and BIC, respectively, as $\tilde{\Delta} \rightarrow 0$. Moreover, $\widetilde{AIC} = AIC + O(\tilde{\Delta}^{1/2})$, and $\widetilde{BIC} = BIC + O(\tilde{\Delta}^{1/2})$.*

Corollary 2 ensures that \widetilde{AIC} and \widetilde{BIC} can be used to approximate the original value of AIC and BIC, and the error vanishes along with the maximum size of data blocks. Both SMR and RASMR algorithms can be used to generate \widetilde{AIC} and \widetilde{BIC} , but numerical experiments overwhelmingly favor RASMR as it improves both the parameter estimates and the representatives.

3.2. Link Function Selection

For GLM (1), the link function g plays a key role in modeling the relationship between a linear combination of predictors and the expectation of the response variable. It is critical for data analysis to choose the most appropriate link function for a given dataset. An information-based approach, such as AIC or BIC, requires precise estimation of the maximum likelihood. Theorem 2 and Corollary 2 provide theoretical justifications for using \widetilde{AIC} or \widetilde{BIC} to select the most appropriate link function.

In our simulation studies (see Section S3 in the Supplementary Materials), SMR fails to choose the logit link function from its competitors (see Table S1), due to the low quality of its representatives. The RASMR with the delta ratio split (Algorithm 2) outperforms SMR with representatives of better quality, making the corresponding \widetilde{AIC} and \widetilde{BIC} more reliable (see Table S2).

3.3. Variable Selection

Variable selection is an essential step in statistical data analysis, including subset selection and stepwise selection (see, for example, [55]).

RASMR can be directly extended for subset selection in big data variable selection problems with a moderate number p of predictors. In the steps involving model fitting and evaluating information criteria (AIC or BIC), RASMR can be implemented readily on the dataset to draw accurate results (see Section S5.1 in the Supplementary Materials).

To balance the processing time and performance of variable selection, we introduce a quick variable screening process using the mean representatives (MR, see Section S4 in the Supplementary Materials). The information criteria \widetilde{AIC} and \widetilde{BIC} based on MR perform sufficiently well (see Table S1).

For forward stepwise variable selection, we recommend using RASMR to perform a finer selection until the stopping criterion is met (see Section S4 in the Supplementary Materials).

3.4. Cross-Validation

Cross-validation is a widely used data-driven technique for evaluating model performance (see, for example, [55]). For big data analysis under GLMs, we extend RASMR for V -fold cross-validation (VFCV) in the big data analysis as follows:

1. Data $\{(X_i, y_i), i = 1, \dots, N\}$ are given with a partition I_1, \dots, I_K of $\{1, \dots, N\}$.

2. A random partition A_1, \dots, A_V of $\{1, \dots, N\}$ is given for V -fold cross-validation.
3. For $j = 1, \dots, V$, fit the target model on the training set $\{(X_i, y_i), i \notin A_j\}$ using RASMR with blocks $I_1 \setminus A_j, \dots, I_K \setminus A_j$ after removing empty ones, and then calculate the aggregated prediction errors $\hat{\mathcal{R}}_j^{VFCV}$ when applying the fitted model on the testing set $\{(X_i, y_i), i \in A_j\}$.
4. Report $N^{-1} \sum_{j=1}^V \hat{\mathcal{R}}_j^{VFCV}$ as the estimated average predictor error.

Note that we do not need to run the same number of iterations on each model fitting. Since RASMR is an iterative approach that can benefit from good initial values of parameter estimates, the estimate from the previous model fitting can be used as the initial values for the next model fitting procedure. Due to this reason, we may set a larger number of iterations for the first RASMR model fitting and then use a smaller number of iterations for later model fitting (see Section S5.2 in the Supplementary Materials).

4. Simulation Studies and Numerical Justifications

4.1. Simulation Setup and Evaluation Method

To justify the performance improvement of the RASMR algorithm on data of massive sizes, we conduct extensive simulations on a main-effects GLM as follows:

$$E(Y_i) = \mu_i \text{ and } \eta_i = g(\mu_i) = \mathbf{X}_i^T \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \dots + \beta_d x_{id}, \tag{8}$$

where $i = 1, \dots, N$. Following [43,56], we choose $d = 7$, $\beta_0 = 0$, and $\beta_1 = \dots = \beta_7 = 0.5$. The feature variables $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ are randomly generated from one of seven different distributions, namely *mzNormal*, *nzNormal*, *ueNormal*, *mixNormal*, T_3 , *EXP*, and *BETA* (see Section S5 in the Supplementary Materials for more details).

For Bernoulli regression models (see Example 1 in Section 4.2), we consider four commonly used link functions, namely *logit*, *cloglog*, *probit*, and *cauchit*. We first simulate $N = 10^6$ data points from each of the seven distributions, respectively, under the *logit* link. We then obtain the data blocks I_1, \dots, I_K using K -means clustering with $K = 1000$. By assuming each of the four link functions, we obtain the corresponding parameter estimates using the MR, SMR, and RASMR algorithms, respectively. We then compare the estimate $\tilde{\beta}_i$ with the full data estimate $\hat{\beta}_i$ under the corresponding link function. The performance of estimation is measured by the root mean square error (RMSE, that is, $[\sum_{i=1}^7 (\tilde{\beta}_i - \hat{\beta}_i)^2 / 7]^{1/2}$). A smaller RMSE indicates a better approach.

Similarly, we consider a Poisson regression model with a *log* link (see Example 2 in Section 4.2) and a Gamma regression model with a *reciprocal* link (see Example 3 in Section 4.2). For each example in Sections 4.2 and 4.3, the corresponding simulation is repeated 100 times. The summarized results, including the average RMSE and the sample standard deviation (std in parenthesis) of 100 RMSEs, are listed in the corresponding tables.

Note that the K -means clustering algorithm performed for each simulation study is used for illustration purposes. It generates a partition with blocks whose data points are homogeneous. Other clustering methods, such as k -medoids (see, for example, [57] and references therein), may be used for the same purpose as well. Our goal here is to evaluate the performance of MR, SMR, and RASMR given the same partition, not to compare different clustering algorithms.

For SMR, each simulation study recommends a number of iterations, T , set to 3, as suggested by [43], as improvement beyond the third iteration is often negligible (see Figure S12 in the Supplementary Materials). In contrast, RASMR typically continues to improve its parameter estimates as T increases, provided the number of data blocks is sufficiently large (see Figure S12). Overall, we recommend setting T to 10 for RASMR algorithms to balance estimation accuracy with computational time. It should be noted that RASMR still outperforms SMR even with the same number of iterations (see Table 6).

4.2. Performance of RASMR, Algorithm 1

In this section, we evaluate the performance of Algorithm 1 on Bernoulli, Poisson, and gamma regression models, respectively.

Example 1. In this example, we consider Bernoulli regression models (see (8)) for binary classifications, with one of four commonly used link functions, namely logit, cloglog, probit, and cauchit. The goal is to check not only the estimation performance of RASMR when the link function is correctly specified but also its performance when the link function is misspecified, which is crucial when selecting the link function.

The results of the corresponding simulation study as described in Section 4.1 are summarized in Table 2. In almost all scenarios, the RMSEs of RASMR are significantly lower than those of the mean representative (MR) approach or the original SMR approach, which implies that the RASMR algorithm (Algorithm 1) outperforms MR and the original SMR algorithm in this example. The estimates based on RASMR are not only much more accurate than the estimates using MR or SMR but also more robust.

Nevertheless, for the case with the cauchit link and nzNormal distribution, the RASMR algorithm performs as poorly as other methods. In Section 4.3, we will show further improvement by using Algorithm 2.

Table 2. Average (std) of RMSEs (10^{-3}) over 100 simulations for binary classification.

Simulation		Binary Classification, K-Means (K = 1000), True Link Function = Logit											
Representatives		MR				Original SMR				RASMR			
Setup	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	
mzNormal	17.9 (0.3)	9.2 (0.3)	9.2 (0.2)	33.4 (1.2)	1.8 (0.6)	1.6 (0.5)	1.0 (0.3)	2.4 (2.4)	3.7×10^{-5} (1.0×10^{-5})	2.3×10^{-1} (1.8×10^{-1})	6.4×10^{-5} (2.6×10^{-5})	1.0×10^{-2} (4.2×10^{-3})	
nzNormal	14.3 (0.7)	5.0 (0.3)	7.0 (0.3)	135 (45.8)	4.0 (1.1)	0.8 (0.4)	1.5 (0.5)	51.6 (34.6)	1.5×10^{-3} (5.0×10^{-4})	2.5×10^{-2} (3.0×10^{-2})	2.5×10^{-4} (7.4×10^{-5})	69.6 (20.0)	
ueNormal	211 (1.4)	114 (1.5)	110 (0.7)	455 (5.2)	3.3 (1.4)	11 (3.4)	2.1 (1.1)	19.4 (10.1)	7.2×10^{-5} (3.3×10^{-6})	4.3 (1.2)	4.7×10^{-4} (9.9×10^{-5})	10.4 (10.2)	
mixNormal	17.5 (0.4)	8.6 (0.6)	8.6 (0.2)	48.1 (3.1)	3.0 (0.9)	1.8 (0.7)	1.2 (0.3)	3.2 (2.5)	3.6×10^{-4} (1.6×10^{-4})	5.0×10^{-1} (1.8×10^{-1})	1.6×10^{-4} (1.8×10^{-4})	1.9×10^{-1} (5.5×10^{-2})	
T ₃	12.2 (3.1)	10.1 (2.6)	7.8 (1.9)	10.0 (2.7)	10.7 (3.1)	15.0 (39.5)	6.7 (1.9)	8.6 (2.7)	6.3×10^{-2} (3.5×10^{-2})	1.2×10^{-1} (7.2×10^{-2})	5.2×10^{-2} (5.2×10^{-2})	2.8×10^{-2} (2.8×10^{-2})	
EXP	12.4 (0.9)	3.9 (0.5)	6.2 (0.5)	10.4 (1.4)	5.8 (1.0)	1.4 (0.3)	2.8 (0.5)	4.5 (1.4)	1.4×10^{-6} (3.3×10^{-7})	3.3×10^{-4} (8.1×10^{-5})	5.3×10^{-6} (6.9×10^{-7})	9.6×10^{-3} (4.4×10^{-3})	
BETA	3.1 (0.8)	1.3 (0.4)	1.7 (0.5)	7.6 (1.7)	2.0 (0.7)	0.9 (0.3)	1.0 (0.3)	11.3 (2.3)	9.2×10^{-7} (6.2×10^{-7})	4.7×10^{-5} (2.4×10^{-5})	2.7×10^{-5} (2.6×10^{-6})	5.6×10^{-3} (6.7×10^{-3})	

Example 2. In this example, we consider Poisson regression with its canonical link function, namely, log link. The corresponding simulation is repeated for 100 times and the results are summarized in Table 3.

From Table 3 we can see that (1) In terms of the average RMSE (the smaller, the better), the original SMR outperforms MR, and RASMR further improves the accuracy of estimates impressively, although not as accurate as the full data estimate in this case (see Table 3 in [43]); (2) In terms of the percentage of NAs (the smaller, the more robust) out of 100 simulations, both MR and RASMR achieve 0% and outperform the original SMR.

Example 3. In the example, we consider Gamma regression with its canonical link function, namely, reciprocal link. Since Gamma regression with reciprocal is relatively fragile on the distribution of features, we only consider features generated from Beta distribution. Again, the data clusters are generated from K-means clustering with K = 1000. Similar to Examples 1 and 2, we obtain the RMSE of the MR, SMR, and RASMR estimates from the full data estimate and summarize the results in Table 4. The pattern of the results is similar to the ones in Table 3, which confirms the significant improvement of RASMR against both MR and SMR under Gamma regression models.

Table 3. Average (std) of RMSEs (10^{-3}) over 100 simulations for the Poisson regression.

Simulation		Poisson Regression, K-Means (K = 1000)				
Representatives	MR	Percentage of NAs	Original SMR	Percentage of NAs	RASMR	Percentage of NAs
mzNormal	37.5 (12.9)	0%	13.2 (10.2)	0%	2.0 (0.5)	0%
nzNormal	37.5 (12.9)	0%	23.6 (15.7)	3%	0.2 (3.6×10^{-2})	0%
ueNormal	82.5 (25.9)	0%	9.5 (7.2)	69%	8.5×10^{-2} (2.3×10^{-2})	0%
mixNormal	49.9 (20.7)	0%	29.5 (17.7)	1%	0.7 (0.1)	0%
T_3	298 (458)	0%	97.2 (144)	10%	31.3 (79.1)	0%
EXP	31.2 (0.7)	0%	6.2 (1.0)	0%	1.9×10^{-4} (5.8×10^{-5})	0%
BETA	0.5 (0.2)	0%	2.3 (0.3)	0%	1.2×10^{-7} (1.8×10^{-9})	0%

Table 4. Average (std) of RMSEs (10^{-3}) over 100 Simulations for Gamma Regression.

Simulation		Gamma Regression, K-Means (K = 1000)				
Representatives	MR	Percentage of NAs	Original SMR	Percentage of NAs	RASMR	Percentage of NAs
Beta	7.4 (0.7)	0%	5.8 (1.2)	11%	2.0×10^{-2} (0.2)	0%

In [43], extensive comparisons between SMR and other big data approaches have been made in terms of computing speed. According to [43], given a partition of data, SMR is noticeably faster than a divide-and-conquer (DC) approach proposed by [5], but slower than the subsampling approach under A-optimality [56] for logistic regression. Using their results of computing time comparison as a reference, we compare the computing speed of RASMR with SMR for binary classification with four link functions: logit, cloglog, probit, and cauchit. The covariates are generated from the seven distributions described previously and the computation is conducted on a PC running Windows 10 Home (Version 2004) with 2.80 GHz 4-core Intel i7-7700HQ and 16 GB of memory. The simulation is repeated 100 times and the average CPU time for one iteration of SMR and RASMR is recorded in Table 5.

Table 5. Average CPU time (seconds) over 100 simulations for binary classification.

Simulation	Binary Classification, K-Means (K = 1000), True Link Function = Logit								
	Representatives	SMR (T = 3)				RASMR (T = 3)			
	Setup	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit
	mzNormal	6.04	8.95	11.09	6.51	7.32	9.75	12.95	7.63
	nzNormal	5.99	8.90	12.82	6.22	6.60	9.82	13.65	6.70
	ueNormal	6.75	10.33	13.55	7.14	7.68	10.38	15.82	7.79
	mixNormal	5.90	8.52	11.26	6.22	6.63	9.52	12.64	7.02
	T_3	6.32	8.43	8.10	6.36	6.77	9.06	8.50	7.00
	EXP	5.66	7.98	10.51	6.24	6.55	9.20	12.11	6.90
	BETA	5.66	7.96	10.73	6.36	6.51	9.10	12.57	7.01

According to Table 5, RASMR is slightly slower than SMR. It is because additional splits are made on a proportion of data blocks, which result in a larger number of data blocks in RASMR, given the same partition of data. Nevertheless, based on our experience, most of the RASMR split requirements are not fulfilled, thus the resulting final number of data blocks of RASMR is only slightly larger than that of SMR.

Similar to [43], we calculate the corresponding RMSE from the full data estimate $[\sum_{i=1}^7 (\hat{\beta}_i - \hat{\beta}_i)^2 / 7]^{1/2}$ and the RMSE from the true parameter values $[\sum_{i=1}^7 (\hat{\beta}_i - \beta_i)^2 / 7]^{1/2}$, respectively. The simulation follows the same settings as in Example 1, and the comparisons are among MR, SMR with $T = 3$, RASMR with $T = 3$, and DC with $K = 1000$. The results are listed in Table 6.

Table 6. Average (std) of RMSEs (10^{-3}) over 100 simulations from full data estimates or true parameter value.

Simulation		Binary Classification, K-Means ($K = 1000$), True Link Function = Logit								
Benchmark		RMSE from Full Data Estimate				RMSE from True Parameter Value				
Setup	Methods	Representative Approaches			DC	Full Data	Representative Approaches			DC
		MR	SMR ($T = 3$)	RASMR ($T = 3$)			MR	SMR ($T = 3$)	RASMR ($T = 3$)	
	mzNormal	17.98(0.31)	1.92(0.62)	0.054(0.018)	6.93(0.12)	3.72(1.08)	18.40(1.02)	4.17(1.16)	3.72(1.08)	7.90(1.04)
	nzNormal	14.30(0.70)	4.53(1.18)	0.29(0.085)	20.20(0.35)	7.23(2.05)	16.04(1.74)	8.58(2.02)	7.24(2.07)	21.49(1.64)
	ueNormal	211.17(1.44)	3.83(1.57)	0.72(0.018)	13.12(0.26)	2.11(0.82)	211.17(1.35)	4.41(1.77)	2.22(0.84)	13.24(1.24)
	mixNormal	17.37(0.33)	2.79(0.83)	0.14(0.043)	11.20(0.20)	4.96(1.33)	17.94(1.05)	5.91(1.48)	4.96(1.33)	12.09(1.15)
	T_3	12.23(3.13)	11.3(3.23)	1.89(0.74)	12.06(0.34)	16.00(4.43)	20.51(5.44)	20.03(5.54)	16.03(4.51)	19.63(3.76)
	EXP	12.4(9.15)	6.58(0.82)	0.014(0.0013)	16.88(0.31)	6.18(1.67)	14.5(2.18)	9.25(2.02)	6.18(1.67)	18.25(2.30)
	BETA	3.03(0.80)	2.34(0.68)	0.00031(0.000090)	5.92(0.20)	7.49(2.38)	7.89(2.30)	7.73(2.34)	7.49(2.38)	9.31(2.54)

According to Table 6, with the same number of iterations ($T = 3$), RASMR still outperforms MR, SMR, and DC. When the comparison is made with respect to the true parameter values, RASMR performs almost as well as the full data estimate with negligible differences. Recall that the data contain the entire information for estimating the parameters and the RASMR approach achieves its theoretical extreme in terms of estimation accuracy in this case.

4.3. Performance of RASMR with the Delta Ratio Split, Algorithm 2

In Section 2.4, we develop Algorithm 2 for estimating the maximum likelihood better, especially for non-Gaussian covariates or predictors. To justify the improvement of Algorithm 2 against RASMR, in this section, we employ the same simulation setting as described in Section 4.1 (see Example 1). For illustration purposes, the threshold δ_0 for delta ratio split is chosen to be 0.05, 0.1, 0.5, and 1, respectively. We fit the logistic regression model using SMR, RASMR, and RASMR with the delta ratio split and various thresholds. The estimated maximum log-likelihood approaches the full data value as the threshold of the delta ratio decreases, except in the T_3 case, where the improvement is negligible (see Figure S11 in the Supplementary Materials). One reason is that in the case of T_3 , the delta ratios are much smaller than those in other cases (see Figure S1 in the Supplementary Materials), and even 0.05 does not capture a good amount of blocks.

To evaluate how the delta ratio split impacts the accuracy of parameter estimates, we choose $K = 100, 500$, and 1000 for K -means clustering, respectively. With moderately large K , such as $K = 500$ or 1000 , RASMR with a delta ratio split performs roughly the same as the original RASMR (see Figure S12 in the Supplementary Materials), and both RASMR and RASMR with a delta ratio split outperform SMR. Nevertheless, when K is as small as 100, the RASMR algorithms may fail for some data structures, such as nzNormal (where the data have imbalanced responses), T_3 (where the data distribute with heavy tails), or mixNormal (where the data come from a mixture of two distributions).

In Table 7, we list the performance of RASMR with the delta ratio split and $\delta_0 = 0.1$ when the link function is misspecified. Comparing Table 7 with Table 2, we can see that RASMR with the delta ratio split (Algorithm 2) is, in general, better than or comparable with Algorithm 1 in terms of estimation accuracy. The improvements are significant in the unbalanced or heavy-tailed cases, such as nzNormal with cauchit link, and ueNormal with cloglog or cauchit links (see Section S5 in the Supplementary Materials for more details).

Table 7. Average (std) of RMSEs (10^{-3}) over 100 simulations using RASMR with the delta ratio split.

Simulation	K-Means ($K = 1000$), True Link Function = Logit			
Approach	RASMR with the Delta Ratio Split, Threshold = 0.1			
Setup	Logit	Cloglog	Probit	Cauchit
mzNormal	4.0×10^{-5} (9.9×10^{-6})	3.2×10^{-2} (3.7×10^{-2})	6.1×10^{-5} (2.4×10^{-5})	8.2×10^{-3} (3.2×10^{-3})
nzNormal	1.6×10^{-3} (5.6×10^{-4})	1.5×10^{-2} (1.4×10^{-2})	2.7×10^{-4} (8.5×10^{-5})	8.8 (4.5)
ueNormal	2.0×10^{-4} 3.4×10^{-3}	9.9×10^{-2} 9.6×10^{-1}	1.2×10^{-3} 2.4×10^{-3}	2.3×10^{-2} 3.3×10^{-2}
mixNormal	3.8×10^{-4} (1.5×10^{-4})	1.9×10^{-1} (1.4×10^{-1})	1.8×10^{-4} (1.8×10^{-4})	7.3×10^{-2} (3.0×10^{-2})
T_3	6.3×10^{-2} $3.4(\times 10^{-2})$	1.0×10^{-1} (5.6×10^{-2})	5.2×10^{-2} (2.5×10^{-2})	2.7×10^{-2} (1.5×10^{-2})
EXP	2.3×10^{-6} (1.8×10^{-6})	4.2×10^{-4} (9.0×10^{-5})	5.8×10^{-6} (7.7×10^{-7})	1.4×10^{-2} (9.7×10^{-3})
BETA	4.6×10^{-6} (3.9×10^{-7})	7.3×10^{-5} (3.4×10^{-5})	2.6×10^{-5} (3.4×10^{-6})	1.0×10^{-2} (3.8×10^{-3})

5. Real Data Analysis

In this section, we use a real example—the airline on-time performance data—collected from the Bureau of Transportation Statistics (<https://www.transtats.bts.gov/>, accessed on 31 August 2018), to illustrate how RASMR works for analyzing real data with a massive size.

5.1. The Airline On-Time Performance Data

The airline on-time performance data contain detailed information on US domestic flights since October 1987. The original data are saved in individual files labeled by month. For illustration purposes, in this study, we use data from between October 1987 and August 2018, 371 files/months in total, which contain 182,751,882 flights. After removing the records with missing or invalid inputs, $N = 179,528,198$ is actually considered in this study.

Following [43], we formulate the data into a binary classification problem aiming to model the flight delay status, a binary response ArrDelayLabel (arrival delay label), which is generated by cutting the continuous variable ARRIVAL DELAY at the 15-min point. For simplicity, the departure time block DepTimeBlk, originally a factor of 17 levels, is regrouped into a factor with 4 levels (1 for 12:00 a.m.–05:59 a.m., 2 for 06:00 a.m.–11:59 a.m., 3 for 12:00 p.m.–05:59 p.m., and 4 for 06:00 p.m.–11:59 p.m.). Moreover, three categorical variables (QUARTER, DayOfWeek, and DepTimeBlk) and three continuous variables (DISTANCE, DepDelay, and CRSTimeElapsed) are used as illustrations to explain the status of flight arrival delays (see Table S8 in the Supplementary Materials for a list of the variables).

5.2. Model Selection

In this section, we select the most appropriate model for the airline data, which consists of 371 months and 179,528,198 flight records.

As described in Section 3.2, we first perform the link function selection using \widetilde{AIC} and five-fold cross-validation based on RASMR with the delta ratio split (Algorithm 2). In this step, all variables listed in Table S8 are included, and the candidate link functions include logit, cloglog, probit, and cauchit.

According to the link function selection results listed in Table 8, we select the logit link for our model (see Figure S14 in the Supplementary Materials for the fitted model based on “glm2” function in [58] and the RASMR representative data).

Table 8. Link function selection using \widetilde{AIC} and 5-fold cross-validation with RASMR.

Link Function	Logit	Cloglog	Probit	Cauchit
\widetilde{AIC}	90284022	97172503	95844750	113052242
5-fold CV with Cross-entropy Loss	1.38441	1.7491	1.8228	2.1535

As described in Section 3.3, we further perform the variable selection for the airline data by implementing the stepwise selection strategy with an initial screening based on MR and fine selection using RASMR (see Section S4 in the Supplementary Materials). The selection criterion is \widetilde{AIC} again. When using “glm2”, we estimate the dispersion parameter by Pearson’s Chi-square statistic divided by its degree of freedom (see Figure S15 in the Supplementary Materials). If n , N , and p represent the number of representatives, the full data size, and the number of parameters in the model, respectively, the degrees of freedom for Pearson’s Chi-square test are $n - p$ for RASMR and $N - p$ for the full data. To estimate the dispersion parameter of full data, the RASMR dispersion parameter estimate needs to be adjusted by the degrees of freedom, i.e., $\hat{\phi} = \hat{\phi}_{RASMR} \cdot (n - p) / (N - p)$, where $\hat{\phi}$ is the dispersion parameter estimated from the full data and $\hat{\phi}_{RASMR}$ is the estimate from the RASMR representative dataset. After such an adjustment, the estimated dispersion parameters are 3291.14 and 3341.67 for the model with all variables and the model with the selected variables, respectively, which indicates strong over-dispersion.

In conclusion, the GLM after variable selection contains only one variable, namely, DepDelay (departure delay in minutes), which is the most informative variable for predicting the arrival delay of a flight.

5.3. Comparison Analysis

For performance evaluation and comparison purposes, we create an oracle model by fitting a logistic regression model using data from between March 2012 and February 2017, 60 months in total, utilizing all explanatory variables listed in Table S8. Then, the oracle response of ArrDelayLabel for the entire dataset is generated accordingly based on the oracle model.

Assuming that we do not know the true link (i.e., logit), we fit the Bernoulli model with one of four different links (namely, logit, cloglog, probit, and cauchit) using four approaches: (1) iterative reweighted least square (IRLS) on the combined data; (2) the mean representative (MR) approach; (3) the score-matching representative (SMR) approach; and (4) RASMR with the delta ratio split and exponential learning rate decay.

The data blocks for MR, SMR, and RASMR are created by the following strategy. The original airline data are saved in monthly files, which are regarded as natural data blocks. Before applying MR, SMR, or RASMR, we further split each of the monthly files according to the distinct values of three categorical variables, namely, QUARTER, DayOfWeek, and DepTimeBlk. Then, each monthly file is divided into $4 \times 7 \times 4 = 112$ sub-blocks. We then split each sub-block further according to the three continuous variables via one of the following two schemes. The first scheme is a correlation-based quantile split. That is, we split each data block at the 25%, 50%, and 75% quantiles of the three continuous variables as the cutting points, and obtain $4 \times 4 \times 4 = 64$ sub-blocks. The second scheme is to apply K -means clustering to the three continuous variables with $K = 64$. As a result, each monthly file is divided into $112 \times 64 = 7168$ sub-blocks (see Section S1 in the Supplementary Materials for more discussion).

We compare MR, SMR, and RASMR with four different data sizes, namely, 60 months, 120 months, 240 months, and 371 months. The IRLS estimates on the combined data are referred to as the full data estimate, but only available for 60-month and 120-month datasets. As a result, in Tables 9 and 10, we list RMSE of MR, SMR, and RASMR (with the delta ratio split) from the IRLS estimates for 60 months and 120 months, and their RMSE from the oracle parameter values for 240 months and 371 months.

Table 9. Average (std) of RMSEs (10^{-3}) over 10 simulations of the airline on-time performance data using K -means clustering.

Simulation		Binary Classification, K -Means ($K = 64$), True Link Function = Logit											
Representatives		MR				SMR				RASMR			
Setup	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	
60 months	28.4	44.3	12.1	135.7	29.3	212.3 (NA removed)	12.6	115.5	1.6	3.0	0.3	34.3	
	(4.5×10^{-4})	(4.6×10^{-5})	(1.2×10^{-4})	(7.0×10^{-4})	(2.2×10^{-3})	8.7 (NA removed)	(7.6×10^{-4})	(1.6×10^{-2})	(1.0×10^{-5})	(8.5×10^{-6})	(2.0×10^{-6})	(2.5×10^{-5})	
120 months	28.3	44.3	12.2	135.4	29.3	212.1 (NA removed)	12.6	115.3	1.6	3.0	0.3	34.3	
	(4.5×10^{-4})	(4.5×10^{-5})	(1.2×10^{-4})	(7.0×10^{-4})	(2.2×10^{-3})	8.6 (NA removed)	(7.6×10^{-4})	(1.6×10^{-2})	(1.0×10^{-5})	(8.5×10^{-6})	(2.0×10^{-6})	(2.5×10^{-5})	
240 months	24.7	41.8	9.0	111.4	33.2	219.8 (NA removed)	12.7	142.0	1.6	3.2	0.3	31.1	
	(3.3×10^{-4})	(2.6×10^{-5})	(8.9×10^{-5})	(7.4×10^{-4})	(4.3×10^{-3})	9.4 (NA removed)	(7.6×10^{-4})	(2.2×10^{-2})	(9.7×10^{-6})	(1.0×10^{-5})	(2.8×10^{-6})	(1.7×10^{-5})	
371 months	24.2	41.4	9.0	111.2	35.9	216.1 (NA removed)	12.7	140.3	1.6	3.2	0.3	31.2	
	(3.1×10^{-4})	(2.6×10^{-5})	(8.9×10^{-5})	(7.4×10^{-4})	(6.2×10^{-3})	9.4 (NA removed)	(7.6×10^{-4})	(2.0×10^{-2})	(9.7×10^{-6})	(1.0×10^{-5})	(2.8×10^{-6})	(1.6×10^{-5})	

Notes: RMSEs of 60 months and 120 months are calculated with respect to the IRLS estimates; RMSEs of 240 months and 371 months are calculated with respect to the oracle parameter values.

Table 10. Average (std) of RMSEs (10^{-3}) over 10 simulations of the airline on-time performance data using correlation-Based quantile split

Simulation		Binary Classification, Correlation-based Quantile Split, True Link Function = Logit											
Representatives		MR				SMR				RASMR			
Setup	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	Logit	Cloglog	Probit	Cauchit	
60 months	62.7	182.8	30.1	376.4	150.1	239.2	49.1	384.0	21.3	20.6	2.4	37.2	
	(1.9×10^{-3})	(7.9×10^{-2})	(1.5×10^{-2})	(4.2×10^{-2})	0.7	0.4	(2.6×10^{-2})	1.5	(1.5×10^{-4})	(1.1×10^{-3})	(1.2×10^{-4})	(6.6×10^{-3})	
120 months	62.5	182.7	30.1	374.2	150.0	239.5	49.4	384.7	21.3	20.6	2.4	37.2	
	(1.9×10^{-3})	(7.9×10^{-2})	(1.5×10^{-2})	(4.2×10^{-2})	0.7	0.4	(2.6×10^{-2})	1.5	(1.5×10^{-4})	(1.1×10^{-3})	(1.2×10^{-4})	(6.6×10^{-3})	
240 months	60.4	180.4	27.7	385.4	147.3	242.1	48.4	373.5	20.2	20.8	2.4	36.9	
	(1.1×10^{-3})	(8.1×10^{-2})	(1.0×10^{-2})	(4.4×10^{-2})	0.7	0.4	(2.7×10^{-2})	1.4	(1.4×10^{-4})	(1.1×10^{-3})	(1.2×10^{-4})	(6.4×10^{-3})	
371 months	60.3	180.1	27.7	385.2	147.3	242.1	48.4	373.4	20.2	20.84	2.4	36.9	
	(1.1×10^{-3})	(8.1×10^{-2})	(1.0×10^{-2})	(4.4×10^{-2})	0.7	0.4	(2.6×10^{-2})	1.4	(1.4×10^{-4})	(1.1×10^{-3})	(1.2×10^{-4})	(6.4×10^{-3})	

Notes: RMSEs of 60 months and 120 months are calculated with respect to the IRLS estimates; RMSEs of 240 months and 371 months are calculated with respect to the oracle parameter values.

According to Tables 9 and 10, RASMR produces consistently more accurate estimates, regardless of the sample size or clustering scheme.

To further compare the convergence properties of SMR and RASMR under different link functions, we plot $\log(\text{RMSE})$ based on SMR or RASMR with K -means clustering in Figure 2 and quantile split clustering in Figure 3, respectively, against the number of iterations. Both Figures 2 and 3 show that RASMR has a better convergence property than the original SMR in this case, which makes RASMR benefit considerably from the increasing number of iterations.

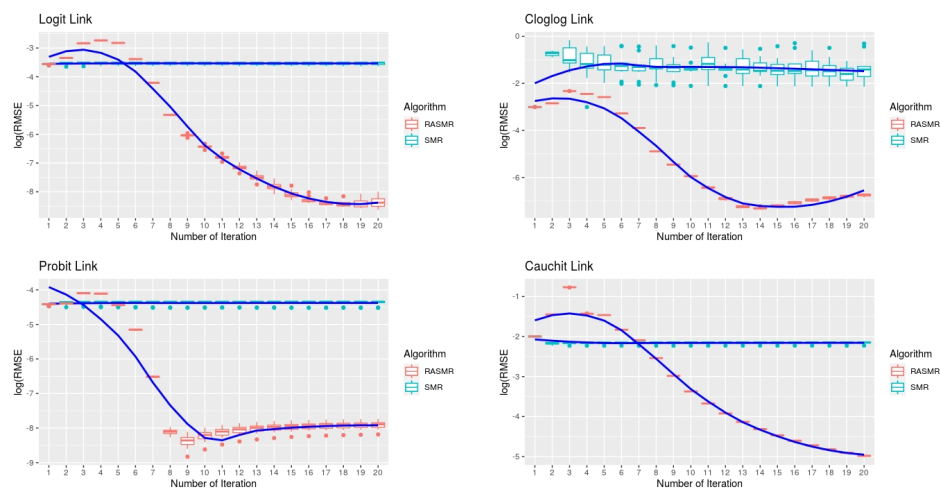


Figure 2. Trend of $\log(\text{RMSE})$ for SMR and RASMR under different link functions for the first 60 months with K -means clustering.

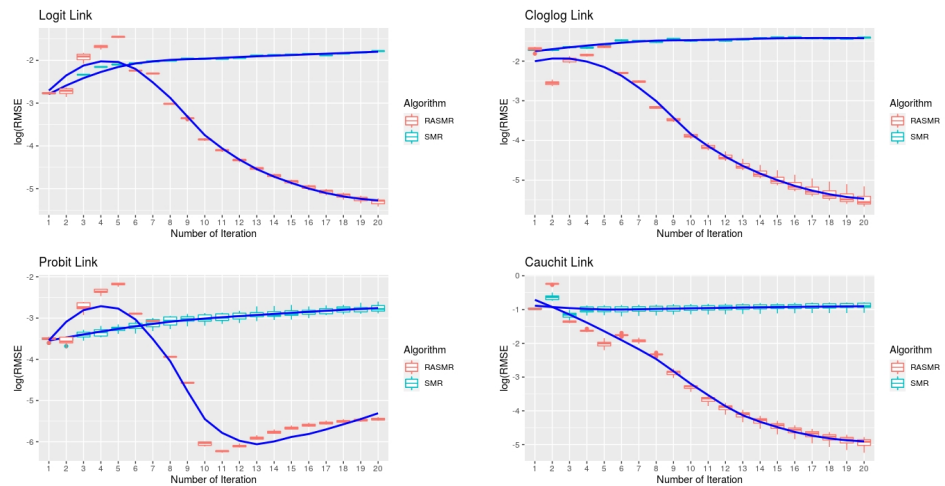


Figure 3. Trend of log(RMSE) for SMR and RASMR estimations under different link functions for the first 60 months with quantile split clustering.

6. Conclusions

The proposed RASMR approach manages to generate more accurate estimates of parameters due to better-quality representatives after additional model-specific splitting. The estimation accuracy improves along with the increasing number of iterations. For data with a relatively simple structure, typically a small number of iterations can provide accurate enough estimates for further data analysis, including model selection. For data with a more complicated structure or a larger number of parameters, we recommend an increased number of iterations and an exponential learning rate decay to ensure better convergence behavior.

For a practical implementation of RASMR, K -means clustering may consume a considerable amount of time. When a natural partition is not available, we recommend small K -means clustering or correlation-based quantile split for a faster clustering process (see Section S1 in the Supplementary Materials for more details). When the number of covariates or predictors is large, we recommend the correlation-based quantile split strategy due to its fast speed.

When applying RASMR to the model selection, we recommend RASMR with the delta ratio split. A default threshold of delta ratio is 1 since the delta ratios rarely go beyond 1 for typical applications. When the competing models have very close performance, we recommend a smaller threshold, such as 0.1, to ensure better-quality representatives.

A key step with RASMR is to construct a good set of representatives and find an accurate approximation $\hat{\beta}$ for the MLE $\hat{\beta}$ based on the full data. When some collinearity exists among the covariates or predictors, $\hat{\beta}$ may not be unique and the RASMR algorithm may encounter an identifiability issue. It is worth exploring how to adjust RASMR for variable selection under the presence of collinearity.

The proposed RASMR approach is especially useful for analyzing big data with binary responses, such as intrusion detection for cyber security [59] and fraud detection for insurance companies [60]. Nevertheless, when the responses have three or more categories, the data may be modeled by multinomial logistic models [61–63] instead of GLMs. In this case, it is challenging to construct representatives for given data blocks that are more efficient than the corresponding mean representatives.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/a17100456/s1>, including more discussions on S1. Generating Data Blocks; S2. More on Delta Ratio Split; S3. More on Link Function Selection; S4. More on Variable Selection Using RASMR; S5. More Simulation Studies; S6. Proofs and Relevant Lemmas; S7. More on Airline Data; S8. More Figures [64–73].

Author Contributions: Conceptualization, D.Z., K.L. and J.Y.; methodology, D.Z. and J.Y.; software, D.Z. and K.L.; validation, D.Z.; formal analysis, D.Z.; investigation, D.Z. and J.Y.; resources, K.L. and J.Y.; data curation, D.Z.; writing—original draft preparation, D.Z. and J.Y.; writing—review and editing, K.L. and J.Y.; visualization, D.Z.; supervision, J.Y.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the U.S. NSF grant DMS-1924859.

Data Availability Statement: The airline on-time performance data are publicly available via the Bureau of Transportation Statistics (<https://www.transtats.bts.gov/> (accessed on 31 August 2018)).

Conflicts of Interest: Author D.Z.’s work was performed while at the University of Illinois at Chicago. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Theorems on Solving the Score-Matching Equation

In this appendix, we provide theorems and remarks on solving the score-matching Equation (3), including explicit solutions for the normal/linear model (Theorem A2), the gamma model (Theorem A5), and the inverse Gaussian model (Theorem A6). With the suggested second split based on $y_i > G(\eta_i)$ and further split for Bernoulli models (Remark A2) and Poisson model (Remark A3), we largely resolve the issue of multiple solutions to (3) in the original SMR algorithm, as explained in Section 2.2.

Theorem A1. Suppose both $v(\eta)$ and $G(\eta)$ are continuous on $[\eta_k^\wedge, \eta_k^\vee]$. If $S(\eta)$ is strictly monotone on $[\eta_k^\wedge, \eta_k^\vee]$, then there exists a unique $\eta_* \in [\eta_k^\wedge, \eta_k^\vee]$ that solves (3).

Theorem A2. For the normal model with an identity link, that is, the usual linear model, there are up to two solutions solving (3):

$$\tilde{\eta}_{k,1} = \frac{1}{2} \left(\tilde{y}_k - \sqrt{\tilde{y}_k^2 - 4\bar{S}} \right), \quad \tilde{\eta}_{k,2} = \frac{1}{2} \left(\tilde{y}_k + \sqrt{\tilde{y}_k^2 - 4\bar{S}} \right), \quad (A1)$$

where $\bar{S} = \bar{\eta}_k \tilde{y}_k - n_k^{-1} \sum_{i \in I_k} \eta_i^2$ and $\bar{\eta}_k = n_k^{-1} \sum_{i \in I_k} \eta_i$. Furthermore,

- (1) If $\eta_i < 0$ and $y_i > G(\eta_i)$ for all $i \in I_k$, then the only solution is $\tilde{\eta}_{k,1}$.
- (2) If $\eta_i > 0$ and $y_i < G(\eta_i)$ for all $i \in I_k$, then the only solution is $\tilde{\eta}_{k,2}$.

In general, if $\tilde{y}_k/2 \notin (\eta_k^\wedge, \eta_k^\vee)$, then the solution is unique.

Note that the essentially same explicit solutions as in (A1) are described in Section 4.2.2.1 of [64].

Along with Theorem A1 and Lemmas S1–S5 in the Supplementary Materials (see Section S6), we conclude the following summarized results for Bernoulli models.

Theorem A3. For Bernoulli models with logit, probit, cloglog, loglog, or cauchit links, a data block I_k satisfying one of the following six conditions yields a unique solution solving (3):

- (i) $\eta_i < \eta_l$ and $y_i < G(\eta_i)$ for all $i \in I_k$;
- (ii) $\eta_l < \eta_i < 0$ and $y_i < G(\eta_i)$ for all $i \in I_k$;
- (iii) $\eta_i > 0$ and $y_i < G(\eta_i)$ for all $i \in I_k$;
- (iv) $\eta_i < 0$ and $y_i > G(\eta_i)$ for all $i \in I_k$;
- (v) $0 < \eta_i < \eta_r$ and $y_i > G(\eta_i)$ for all $i \in I_k$;
- (vi) $\eta_i > \eta_r$ and $y_i > G(\eta_i)$ for all $i \in I_k$,

where $\eta_l < 0$ and $\eta_r > 0$ are constants listed in Table 1. A general data block under Bernoulli models may yield up to two solutions solving (3).

Remark A1. In practice, the “<” and “>” in Theorem A3 can be relaxed to “≤” and “≥”, respectively, as long as “=” can only be attained by a small portion of the observations.

Remark A2. Theorem A3 suggests further split sub-blocks according to η_l or η_r , which are constants associated with link functions. More specifically, if $\eta_i < 0$ and $y_i < G(\eta_i)$ for all $i \in I_k$, we further split I_k into two sub-blocks at the splitting point η_l (see cases (i) and (ii)); if $\eta_i > 0$ and $y_i > G(\eta_i)$ for all $i \in I_k$, we further split I_k at the splitting point η_r (see cases (v) and (vi)). Each resulting block yields a unique solution solving (3), which is equivalent to solving $S(\eta) = \bar{S}$ (see (6)). For example, for Bernoulli models with logit link, that is, logistic regression models, we have the following:

$$S(\eta) = \begin{cases} -\frac{\eta e^\eta}{1+e^\eta}, & \text{for cases (i), (ii) and (iii);} \\ \frac{\eta}{1+e^\eta}, & \text{for cases (iv), (v), and (vi).} \end{cases}$$

Furthermore, the solutions are associated with the Lambert W-function $P(z)$, which solves $z = we^{-w}$ (see R function `lambertW` in package `VGAM` [65]). It can be verified that, for cases (ii) and (iii), $\tilde{\eta}_k = P(-\bar{S}e^{\bar{S}}) - \bar{S}$; for cases (iv) and (v), $\tilde{\eta}_k = \bar{S} - P(-\bar{S}e^{\bar{S}})$. Nevertheless, `lambertW` does not provide solutions for cases (i) and (vi) for now. One may use a quasi-Newton algorithm to solve them.

Theorem A4. For the Poisson model with a log link, suppose either $\eta_i > 0$ for all $i \in I_k$ or $\eta_i < 0$ for all $i \in I_k$. Then, there are up to two solutions $\tilde{\eta}_{k,1} \in [\eta_k^\wedge, u(\tilde{y}_k)]$ and $\tilde{\eta}_{k,2} \in [u(\tilde{y}_k), \eta_k^\vee]$ solving (3), where $u(y) \geq -1$ denotes the unique root of the transcendent equation $e^\eta(1 + \eta) = y$ given $y \geq 0$. Special cases include the following:

- (i) If $y_i = 0$ and $\eta_i \geq 0$ for all $i \in I_k$, then there exists a unique solution $\tilde{\eta}_k = 1 + u(-\bar{S}/e) \geq 0$.
- (ii) If $y_i = 0$ and $\eta_i \in (-1, 0)$ for all $i \in I_k$, then there exists a unique solution in $(-1, 0)$.
- (iii) If $y_i = 0$ and $\eta_i \leq -1$ for all $i \in I_k$, then there exists a unique solution in $(-\infty, -1]$.

Remark A3. Theorem A4 suggests that the observations with $y_i = 0$ be extracted and further partitioned into three sub-blocks according to $\eta_i \in (-\infty, -1]$, $(-1, 0)$ and $[0, \infty)$, respectively, or with splitting points -1 and 0 . Then, $\tilde{\eta}_k$ solving (3) within each sub-block is unique and can be obtained easily. Moreover, all three special cases in Theorem A4 are related to the well-known Lambert W-function $P(z)$, which solves $z = we^w$. The function “`lambertW`” from R package `VGAM` may be used for this purpose as well.

Theorem A5. For the gamma model with a reciprocal link, $\bar{\eta}_k = n_k^{-1} \sum_{i \in I_k} \eta_i$ is the unique solution solving (3). Furthermore, if $y_i < G(\eta_i)$ for all $i \in I_k$, then $\bar{\eta}_k \in (0, \tilde{y}_k^{-1})$; if $y_i > G(\eta_i)$ for all $i \in I_k$, then $\bar{\eta}_k \in (\tilde{y}_k^{-1}, \infty)$.

Remark A4. For the gamma model with a reciprocal link, $\tilde{\eta}_k = \bar{\eta}_k = \bar{\mathbf{X}}_k \boldsymbol{\beta}$ is the only solution solving (3), where $\bar{\mathbf{X}}_k = n_k^{-1} \sum_{i \in I_k} \mathbf{X}_i$ is the mean representative. In this case, $v(\eta)$ is a constant (see Table 1 in [43]), and we have the following:

$$\tilde{\mathbf{X}}_k = \frac{\sum_{i \in I_k} (y_i - \eta_i^{-1}) \mathbf{X}_i}{n_k(\tilde{y}_k - \bar{\eta}^{-1})} \tag{A2}$$

according to (4). That is, $\tilde{\mathbf{X}}_k \neq \bar{\mathbf{X}}_k$ as in (A2), even if $\tilde{\eta}_k = \bar{\mathbf{X}}_k \boldsymbol{\beta}$.

Theorem A6. For the inverse Gaussian model with an inverse-square link, in general, there are up to two solutions, i.e.,

$$\tilde{\eta}_{k,1} = \frac{1 - 4\tilde{y}_k \bar{S} - \sqrt{1 - 8\tilde{y}_k \bar{S}}}{2\tilde{y}_k^2}, \quad \tilde{\eta}_{k,2} = \frac{1 - 4\tilde{y}_k \bar{S} + \sqrt{1 - 8\tilde{y}_k \bar{S}}}{2\tilde{y}_k^2}$$

solving (3), where $\bar{S} = n_k^{-1} \sum_{i \in I_k} \frac{1}{2}(\eta_i^{-1/2} - \tilde{y}_k)\eta_i$, and $\tilde{\eta}_{k,1} \leq (4\tilde{y}_k^2)^{-1} \leq \tilde{\eta}_{k,2}$. Furthermore, if $y_i > G(\eta_i)$ for all $i \in I_k$, then $\bar{S} < 0$ and $\tilde{\eta}_k = (1 - 4\tilde{y}_k \bar{S} + \sqrt{1 - 8\tilde{y}_k \bar{S}})/(2\tilde{y}_k^2)$ is the only solution.

References

1. Dautov, R.; Distefano, S. Quantifying volume, velocity, and variety to support (big) data-intensive application development. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2843–2852.
2. Fan, J.; Han, F.; Liu, H. Challenges of big data analysis. *Natl. Sci. Rev.* **2014**, *1*, 293–314. [[CrossRef](#)] [[PubMed](#)]
3. Wang, C.; Chen, M.; Schifano, E.; Wu, J.; Yan, J. Statistical methods and computing for big data. *Stat. Its Interface* **2016**, *9*, 399–414. [[CrossRef](#)] [[PubMed](#)]
4. Lin, L.; Lu, J. A race-DC in Big Data. *arXiv* **2019**, arXiv:1911.11993.
5. Lin, N.; Xi, R. Aggregated estimating equation estimation. *Stat. Its Interface* **2011**, *4*, 73–83. [[CrossRef](#)]
6. Chen, X.; Xie, M.g. A split-and-conquer approach for analysis of extraordinarily large data. *Stat. Sin.* **2014**, *24*, 1655–1684.
7. Schifano, E.; Wu, J.; Wang, C.; Yan, J.; Chen, M. Online updating of statistical inference in the big datasetting. *Technometrics* **2016**, *58*, 393–403. [[CrossRef](#)]
8. Zhao, T.; Cheng, G.; Liu, H. A partially linear framework for massive heterogeneous data. *Ann. Stat.* **2016**, *44*, 1400–1437. [[CrossRef](#)]
9. Lee, J.D.; Liu, Q.; Sun, Y.; Taylor, J.E. Communication-efficient sparse regression. *J. Mach. Learn. Res.* **2017**, *18*, 115–144.
10. Battey, H.; Fan, J.; Liu, H.; Lu, J.; Zhu, Z. Distributed testing and estimation under sparse high dimensional models. *Ann. Stat.* **2018**, *46*, 1352–1382. [[CrossRef](#)]
11. Shi, C.; Lu, W.; Song, R. A massive data framework for M-estimators with cubic-rate. *J. Am. Stat. Assoc.* **2018**, *113*, 1698–1709. [[CrossRef](#)]
12. Chen, X.; Liu, W.; Zhang, Y. Quantile regression under memory constraint. *Ann. Stat.* **2019**, *47*, 3244–3273. [[CrossRef](#)]
13. Ma, P.; Sun, X. Leveraging for big data regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2015**, *7*, 70–76. [[CrossRef](#)]
14. Wang, H.; Yang, M.; Stufken, J. Information-based optimal subdata selection for big data linear regression. *J. Am. Stat. Assoc.* **2019**, *114*, 393–405. [[CrossRef](#)]
15. Cheng, Q.; Wang, H.; Yang, M. Information-based optimal subdata selection for big data logistic regression. *J. Stat. Plan. Inference* **2020**, *209*, 112–122. [[CrossRef](#)]
16. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
17. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259.
18. Breiman, L. Stacked regressions. *Mach. Learn.* **1996**, *24*, 49–64. [[CrossRef](#)]
19. Bühlmann, P.; Meinshausen, N. Maging: Maximin aggregation for inhomogeneous large-scale data. *Proc. IEEE* **2015**, *104*, 126–135.
20. da Conceição Costa, M.; Macedo, P. Normalized entropy aggregation for inhomogeneous large-scale data. In Proceedings of the Theory and Applications of Time Series Analysis: Selected Contributions from ITISE 2018, Granada, Spain, 19–21 September 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 19–29.
21. Costa, M.C.; Macedo, P.; Cruz, J.P. Neagging: An aggregation procedure based on normalized entropy. In Proceedings of the AIP Conference Proceedings, Rhodes, Greece, 17–23 September 2020; AIP Publishing: Melville, NY, USA, 2022; Volume 2425.
22. Tran, D.; Toulis, P.; Airolidi, E.M. Stochastic gradient descent methods for estimation with large datasets. *arXiv* **2015**, arXiv:1509.06459.
23. Lin, J.; Rosasco, L. Optimal Rates for multi-pass stochastic gradient methods. *J. Mach. Learn. Res.* **2017**, *18*, 1–47.
24. Airolidi, E.; Toulis, P. Stochastic Gradient Methods for Principled Estimation with Large Data Sets. In *Handbook of Big Data*; Chapman & Hall: London, UK, 2016; pp. 243–266.
25. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv* **2016**, arXiv:1610.02527.
26. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; PMLR: New York, NY, USA, 2017; pp. 1273–1282.
27. Stich, S.U. Local SGD converges fast and communicates little. *arXiv* **2018**, arXiv:1805.09767.
28. Stich, S.U.; Karimireddy, S.P. The error-feedback framework: Better rates for SGD with delayed gradients and compressed updates. *J. Mach. Learn. Res.* **2020**, *21*, 1–36.
29. Khaled, A.; Mishchenko, K.; Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 26–28 August 2020; PMLR: New York, NY, USA, 2020; pp. 4519–4529.
30. Spiridonoff, A.; Olshevsky, A.; Paschalidis, Y. Communication-efficient sgd: From local sgd to one-shot averaging. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24313–24326.
31. Wang, J.; Joshi, G. Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. *J. Mach. Learn. Res.* **2021**, *22*, 1–50.
32. Zhou, F.; Cong, G. On the convergence properties of a K -step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv* **2017**, arXiv:1708.01012.
33. Koloskova, A.; Loizou, N.; Boreiri, S.; Jaggi, M.; Stich, S. A unified theory of decentralized sgd with changing topology and local updates. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; PMLR: New York, NY, USA, 2020; pp. 5381–5393.

34. Jiang, P.; Agrawal, G. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), Montréal, QC, Canada, 3–8 December 2018; Curran Associates: Red Hook, NY, USA, 2018; pp. 2530–2541.
35. Haddadpour, F.; Mahdavi, M. On the convergence of local descent methods in federated learning. *arXiv* **2019**, arXiv:1910.14425.
36. Zhu, Z.; Hong, J.; Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 12878–12889.
37. Li, A.; Sun, J.; Li, P.; Pu, Y.; Li, H.; Chen, Y. Hermes: An efficient federated learning framework for heterogeneous mobile clients. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, New Orleans, LA, USA, 31 January–4 February 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 420–437.
38. Sery, T.; Shlezinger, N.; Cohen, K.; Eldar, Y.C. Over-the-air federated learning from heterogeneous data. *IEEE Trans. Signal Process.* **2021**, *69*, 3796–3811. [[CrossRef](#)]
39. Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; Ding, Z.; Chen, C. Local learning matters: Rethinking data heterogeneity in federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8397–8406.
40. Qu, L.; Zhou, Y.; Liang, P.P.; Xia, Y.; Wang, F.; Adeli, E.; Fei-Fei, L.; Rubin, D. Rethinking architecture design for tackling data heterogeneity in federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 10061–10071.
41. Fang, X.; Ye, M. Robust federated learning with noisy and heterogeneous clients. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 10072–10081.
42. Ye, M.; Fang, X.; Du, B.; Yuen, P.C.; Tao, D. Heterogeneous federated learning: State-of-the-art and research challenges. *Acm Comput. Surv.* **2023**, *56*, 1–44. [[CrossRef](#)]
43. Li, K.; Yang, J. Score-matching representative approach for big data analysis with generalized linear models. *Electron. J. Stat.* **2022**, *16*, 592–635. [[CrossRef](#)]
44. Bowman, C. Data localization laws: An emerging global trend. *JURIST–Hotline*, 6 January 2017. Available online: <https://www.jurist.org/commentary/2017/01/courtney-bowman-data-localization/> (accessed on 8 October 2024).
45. Chander, A.; Lê, U.P. Data nationalism. *Emory LJ* **2014**, *64*, 677.
46. McCullagh, P.; Nelder, J. *Generalized Linear Models*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 1989.
47. Dobson, A.; Barnett, A. *An Introduction to Generalized Linear Models*, 4th ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2018.
48. Green, P.J. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J. R. Stat. Soc. Ser.* **1984**, *46*, 149–170. [[CrossRef](#)]
49. Gentle, J.E. *Matrix Algebra*; Springer: Berlin/Heidelberg, Germany, 2007.
50. R Core Team and contributors worldwide. In *The R Stats Package*; R Package Version 4.5.0.; R Foundation for Statistical Computing: Vienna, Austria, 2024.
51. Peng, L.; Kümmerle, C.; Vidal, R. On the convergence of IRLS and its variants in outlier-robust estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 17808–17818.
52. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
53. You, K.; Long, M.; Wang, J.; Jordan, M.I. How does learning rate decay help modern neural networks? *arXiv* **2019**, arXiv:1908.01878.
54. Akaike, H. Information theory and an extension of the maximum likelihood principle. In Proceedings of the 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, 2–8 September 1971; Akademiai Kiado: Budapest, Hungary, 1973; pp. 267–281.
55. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009.
56. Wang, H.; Zhu, R.; Ma, P. Optimal subsampling for large sample logistic regression. *J. Am. Stat. Assoc.* **2018**, *113*, 829–844. [[CrossRef](#)]
57. Schubert, E.; Rousseeuw, P.J. Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms. *Inf. Syst.* **2021**, *101*, 101804. [[CrossRef](#)]
58. Marschner, I.; Donoghoe, M.W. *glm2: Fitting Generalized Linear Models*; R Package Version 1.2.1.; R Foundation for Statistical Computing: Vienna, Austria, 2018.
59. Kumar, V.S. A Big Data Analytical Framework for Intrusion Detection Based on Novel Elephant Herding Optimized Finite Dirichlet Mixture Models. *Int. J. Data Inform. Intell. Comput.* **2023**, *2*, 11–20.
60. Jones, K.I.; Sah, S. The Implementation of Machine Learning In The Insurance Industry with Big Data Analytics. *Int. J. Data Inform. Intell. Comput.* **2023**, *2*, 21–38.
61. Glonek, G.; McCullagh, P. Multivariate logistic models. *J. R. Stat. Soc. Ser.* **1995**, *57*, 533–546. [[CrossRef](#)]
62. Zocchi, S.; Atkinson, A. Optimum experimental designs for multinomial logistic models. *Biometrics* **1999**, *55*, 437–444. [[CrossRef](#)]
63. Bu, X.; Majumdar, D.; Yang, J. D-optimal Designs for Multinomial Logistic Models. *Ann. Stat.* **2020**, *48*, 983–1000. [[CrossRef](#)]

64. Li, K. Score-Matching Representative Approach for Big Data Analysis with Generalized Linear Models. Ph.D. Thesis, University of Illinois at Chicago, Chicago, IL, USA, 2018.
65. Yee, T.; Moler, C. *VGAM: Vector Generalized Linear and Additive Models*; R Package Version 1.1-11.; R Foundation for Statistical Computing: Vienna, Austria, 2024.
66. Lohr, S. *Sampling: Design and Analysis*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019.
67. Gordon, R. Values of Mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. *Ann. Math. Stat.* **1941**, *12*, 364–366. [[CrossRef](#)]
68. Birnbaum, Z. An inequality for Mill's ratio. *Ann. Math. Stat.* **1942**, *13*, 245–246. [[CrossRef](#)]
69. Mitrovic, D.; Vasic, P. *Analytic Inequalities*; Springer: Berlin/Heidelberg, Germany, 1970.
70. Baricz, A. Mills' ratio: Monotonicity patterns and functional inequalities. *J. Math. Anal. Appl.* **2008**, *340*, 1362–1370. [[CrossRef](#)]
71. Marshall, A.; Olkin, I. *Life Distributions*; Springer: Berlin/Heidelberg, Germany, 2007.
72. Corless, R.; Gonnet, G.; Hare, D.; Jeffrey, D.; Knuth, D. On the Lambert W function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [[CrossRef](#)]
73. Dunn, P.; Smyth, G. *Generalized Linear Models with Examples in R*; Springer: Berlin/Heidelberg, Germany, 2018.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.