

RESEARCH ARTICLE OPEN ACCESS

Scalable Inference via Averaged Robbins-Monro Bootstrap

Giuseppe Alfonzetti  | Ruggero Bellio

Università degli Studi di Udine, Udine, Italy

Correspondence: Giuseppe Alfonzetti (giuseppe.alfonzetti@uniud.it)**Received:** 13 December 2024 | **Revised:** 28 July 2025 | **Accepted:** 29 September 2025**Funding:** This work was supported by Università degli Studi di Udine.**Keywords:** bag of little bootstraps | massive data | Ruppert-Polyak averaging | scalability | stochastic approximations

ABSTRACT

Bootstrap procedures represent a straightforward approach to assessing the uncertainty around estimates of interest in statistical models. However, with the rising prevalence of massive datasets in statistical problems, the computational cost of bootstrap methods can quickly become prohibitive in many settings. To this end, this paper proposes the Averaged Robbins-Monro Bootstrap (ARM-B), a scalable tool for estimating parameter variability via multiple chains of Robbins-Monro updates. The method is illustrated in large-scale Poisson regression and logistic regression settings and compared with the alternative scalable method given by the bag of little bootstraps (BLB). Some simulation experiments and an illustrative analysis on a large-scale dataset show that ARM-B has comparable accuracy with ordinary bootstrap, but, at the same time, it is significantly less computationally demanding and quite competitive with BLB.

1 | Introduction

In the last decades, there has been an increased use of sensors and automatic systems to predict and monitor processes in industry. To deal with such a prevalent trend, statistics and machine learning methods had to adapt to datasets of growing dimensions progressively. One widely spread strategy to enhance the computational scalability of model estimation on large datasets has been the use of stochastic optimisation algorithms, which stemmed from the seminal work of [1] and became the de facto standard choice in many large-scale applications in machine learning. We refer to [2] for a comprehensive review.

While the focus of research in the stochastic optimisation area has typically been on pointwise estimates, in recent years, there has been a growing interest in the interpretability and explainability of large-scale models. Thus, more relevance has been given to the possibility of drawing statistical inferences with such estimates. See the works of [3–6] and [7] among many. Much of the

research in this direction builds on the fundamental results of [8], who proved the statistical optimality and asymptotic normality of averaged stochastic estimators.

However, depending on the complexity of the model of interest and the dimensions of the data available, one can argue that there are still plenty of applications where the computational affordability of stochastic methods is not really needed for point estimates and the maximum likelihood estimator (MLE) can be computed numerically in a reasonable amount of time on modern commodity hardware, especially if parallelisation is possible. Nevertheless, there is a large variety of settings where evaluating the MLE once is computationally affordable, but repeating the estimation hundreds of times to assess the quality of the estimates via bootstrap methods [9–11] can be unfeasible due to practical time constraints. Very little research has been carried out in such settings, with the notable exception of the bag of little bootstraps (BLB) proposed in Reference [12].

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Applied Stochastic Models in Business and Industry* published by John Wiley & Sons Ltd.

To deal with such scenarios, we propose the Averaged Robbins-Monro Bootstrap (ARM-B), which is a bootstrapped version of the averaged Robbins-Monro updates studied in Reference [8], and exploits the intrinsic variability of stochastic estimators around the MLE to approximate its asymptotic covariance matrix.

In the following, we illustrate the proposal and provide some theoretical support for it. Then, we compare it to the conventional bootstrap and the BLB approach in two simulation studies. Finally, the method is applied to a real dataset, to obtain inferential results on the parameters of a classification model for the failure of the Air Pressure System (APS) on heavy Scania trucks.¹

2 | Background

Let us consider a sample of dimension n of independently and identically distributed observations $(Y_1, X_1), \dots, (Y_n, X_n)$, with each pair (Y, X) following a certain unspecified distribution \mathbf{F} . We also denote the empirical distribution of the observed sample with \mathbb{F}_n .

We then assume a parametric specification for the conditional distribution of Y given X , given by $p(Y; X, \theta)$, where $\theta \in \mathbb{R}^d$ is the vector of model parameters. The Maximum Likelihood Estimator (MLE), $\hat{\theta}_{MLE} = \hat{\theta}_{MLE}(\mathbb{F}_n)$, is obtained by solving the estimating equation

$$E_{\mathbb{F}_n} \{ \nabla \log p(Y; X, \hat{\theta}_{MLE}) \} = 0, \quad (1)$$

where $\nabla \log p(Y; X, \theta)$ is the gradient of the log-density function with respect to θ . Let $\mathbf{Q}_n(\mathbf{F})$ be the distribution of $\hat{\theta}_{MLE}$ with n observations sampled from \mathbf{F} . Then, to assess the uncertainty around maximum likelihood estimates, we are interested in evaluating their variance, i.e., $V(\mathbf{Q}_n(\mathbf{F}))$. A standard approach is given by the nonparametric bootstrap, for which textbook treatments are given by Efron and Tibshirani [11] and Davison and Hinkley [9], among many others. The bootstrap requires drawing from \mathbb{F}_n further R independent samples, with empirical distributions given by $\mathbb{F}_n^{(1)}, \dots, \mathbb{F}_n^{(R)}$. Then, for the r -th sample, $r = 1, \dots, R$, the MLE given by $\hat{\theta}_{MLE}(\mathbb{F}_n^{(r)})$ is computed. The R maximum likelihood estimates define the empirical distribution $\mathbb{Q}_{n,R}(\mathbb{F}_n)$, which is used to approximate the variance of $\hat{\theta}_{MLE}$ by computing $V(\mathbb{Q}_{n,R}(\mathbb{F}_n))$, i.e., the sample variance of the R realisations of the maximum likelihood estimator with data resampled from \mathbb{F}_n . Basic bootstrap theory (see for example [9, Ch. 2]), ensures that $V(\mathbb{Q}_{n,R}(\mathbb{F}_n))$ converges in probability to $V(\mathbf{Q}_n(\mathbf{F}))$. In addition, the classic result by White [13] guarantees that, under standard regularity conditions but without assuming a correctly specified parametric model, the MLE converges almost surely to θ^* , the parameter vector minimising the Kullback-Leibler distance from the true distribution. In other words, it holds $\hat{\theta}_{MLE} \xrightarrow[n \rightarrow \infty]{a.s.} \theta^*$, and the MLE behaves asymptotically according to

$$\sqrt{n}(\hat{\theta}_{MLE} - \theta^*) \xrightarrow[n \rightarrow \infty]{d} N(0; H^{*-1} J^* H^{*-1}) \quad (2)$$

where $H^* = -E_{\mathbf{F}} \{ \nabla^2 \log p(Y; X, \theta^*) \}$ and $J^* = \text{Var}_{\mathbf{F}} \{ \nabla \log p(Y; X, \theta^*) \}$, with the symbol \xrightarrow{d} denoting convergence in distribution. By combining (2) with the bootstrap theory, we get that, under the regularity conditions outlined in Reference [13],

$V(\mathbb{Q}_{n,R}(\mathbb{F}_n))$ approximates $n^{-1} H^{*-1} J^* H^{*-1}$ when both n and R are large enough.

While the bootstrap computation is straightforward in principle, for large-scale data settings, the computational burden might be demanding. Indeed, the computation of $\hat{\theta}_{MLE}$ requires to solve (1) by iteratively re-evaluating the log-likelihood gradient $E_{\mathbb{F}_n} \{ \nabla \log p(Y; X, \theta) \}$, whose computational complexity is $O(nd)$. Consequently, obtaining the estimation across the R bootstrap samples often involves prohibitive computational times. Reference [12] propose the bag of little bootstraps (BLB), a modified bootstrap procedure that allows for efficient computations with large sample sizes. Consider drawing S subsets from \mathbb{F}_n of dimension $b = n^\gamma$, $0 < \gamma \leq 1$, and let us denote with $\mathbb{F}_{n,b}^{(s)}$ their empirical distributions, with $s = 1, \dots, S$. For each of the subsets, they propose to resample R times n -out-of- b observations, leading to R independent samples with empirical distributions $\mathbb{F}_{n,b}^{(s)(r)}$, with $r = 1, \dots, R$, and compute $\hat{\theta}_{MLE}(\mathbb{F}_{n,b}^{(s)(r)})$ on each of them. The collection of maximum likelihood estimates on the R resamples within the same subset s is characterised by the empirical distribution $\mathbb{Q}_{n,b}(\mathbb{F}_{n,b}^{(s)})$. Thus they propose to estimate $V(\mathbf{Q}_n(\mathbf{F}))$ by computing $S^{-1} \sum_{s=1}^S V(\mathbb{Q}_{n,b}(\mathbb{F}_{n,b}^{(s)}))$. The computational advantage of BLB over the classical bootstrap procedure lies in the evaluation of $\hat{\theta}_{MLE}(\mathbb{F}_{n,b}^{(s)(r)})$. If the model allows for data reduction via observation weights, it is straightforward to notice that, since $b < n$, then $E_{\mathbb{F}_{n,b}^{(s)(r)}} \{ \nabla \log p(Y; X, \theta) \}$ complexity reduces to $O(bd)$ which can be much smaller than the standard $O(nd)$. Thus, with its peculiar resampling procedure, BLB allows for a lower computational burden than standard nonparametric bootstrap when evaluating point estimates, which is its main benefit.

A different area of the literature has focused on reducing the computational cost of $\hat{\theta}_{MLE}(\mathbb{F}_n)$ from a different perspective than data reduction observation weights. It is the case of parameter estimation via stochastic optimisation, which has gained significant traction since the seminal work of Robbins and Monro [1], both in the machine learning and statistics communities. Given a starting point θ_0 and a suitable decreasing scheduling for the stepsize η_t , the simple Robbins-Monro scheme proceeds by progressively updating the parameter estimate as

$$\theta_t = \theta_{t-1} + \eta_t \nabla \log p(Y_t; X_t, \theta_{t-1}), \quad \text{for } t = 1, \dots, \infty. \quad (3)$$

In the update (3), at each iteration t , a new data point (Y_t, X_t) is sampled from some suitable distribution \mathbf{P} . Such an updating rule is computationally appealing since it requires evaluating the log-likelihood gradient at a single data point rather than at the full sample, and thus has an $O(d)$ cost. Under suitable regularity conditions for η_t , $\nabla \log p(Y; X, \theta)$ and $\log p(Y; X, \theta)$, Robbins and Monro [1] show that θ_t converges almost surely to the solution of the equation $E_{\mathbf{P}} \{ \nabla \log p(Y; X, \theta) \} = 0$ as t diverges. It readily follows that by setting $\mathbf{P} \equiv \mathbb{F}_n$, the estimate θ_t converges to $\hat{\theta}_{MLE}(\mathbb{F}_n)$. In practice, while the scheme (3) is computationally very efficient, the values of θ_0 and η_0 play an essential role, affecting both the stability of the updating trajectories and the number of iterations needed for the algorithm to converge.

In the following, we propose to leverage the computational advantage of (3) to lower the estimation burden of bootstrap procedures.

3 | Averaged Robbins-Monro Bootstrap (ARM-B)

The ARM-B estimator is based on the simple idea of taking advantage of the variability of stochastic estimators around the MLE to estimate the asymptotic variability of the MLE around the true parameter value. In this sense, it enters the inferential process only after the practitioner has computed $\hat{\theta}_{MLE}(\mathbb{F}_n)$ on the original data. From a theoretical perspective, the consistency of the ARM-B estimator builds on the standard type of assumptions outlined in Reference [8] and used, possibly with some variations, throughout the literature of averaged stochastic estimators [3–5, 7]. The assumptions are hence as follows.

Assumption 1. The log-likelihood function on the sample is concave, continuously differentiable over θ , and twice continuously differentiable at $\hat{\theta}_{MLE}$. Its gradient, as well as its Hessian at $\hat{\theta}_{MLE}$, are Lipschitz continuous, and its Hessian at $\hat{\theta}_{MLE}$ is negative definite.

Assumption 2. $E\|\nabla \log p(Y; x, \theta)\|^2 \leq C(1 + \|\theta\|^2)$ for some positive C and $E\|\nabla \log p(Y; x, \theta) - \nabla \log p(Y; x, \hat{\theta}_{MLE})\|^2 \leq \delta(\|\theta - \hat{\theta}_{MLE}\|)$ with $\delta(x) \rightarrow 0$ as $x \rightarrow 0$.

Assumption 3. The learning schedule satisfies $\lim_{t \rightarrow \infty} \sum_{s=1}^t \eta_t = \infty$, $\lim_{t \rightarrow \infty} \sum_{s=1}^t \eta_t^2 < \infty$ and $\lim_{t \rightarrow \infty} \sum_{s=1}^t \eta_t / \sqrt{t} < \infty$.

Assumption 1 guarantees the strong convexity of the negative sample log-likelihood and imposes a set of smoothness conditions on its derivatives, while Assumption 2 controls the variance of the stochastic gradients at each iteration. Assumption 3, finally, imposes a decaying and controlled schedule on the learning rate, guaranteeing the convergence of the stochastic updates.

Under assumptions equivalent to 1–3, [8] show that the averaged Robbins-Monro estimate, i.e., the average over $t = 1, \dots, T$ of θ_t specified as in Equation (3), not only converges to the root of $E_{\mathbf{P}}\{\nabla \log p(Y; X, \theta)\} = 0$ as $T \rightarrow \infty$, but it is also asymptotically normally distributed around it. We leverage such results to show that, by running R parallel chains of averaged Robbins-Monro updates, the empirical variance of the obtained R independent pointwise stochastic estimates can be used as an estimator of $V(\mathbf{Q}_n(\mathbf{F}))$ when $\mathbf{P} \equiv \mathbb{F}_n$. Hence, while the stochastic procedure outlined in Reference [8] leads to a pointwise estimator of $\hat{\theta}_{MLE}(\mathbb{F}_n)$, we propose to use its variability around it to estimate the variability of $\hat{\theta}_{MLE}(\mathbb{F}_n)$ around the true parameter value θ^* .

Let Δ_t be the vector of differences with the MLE of the stochastic estimates computed by (3), i.e., $\Delta_t = \theta_t - \hat{\theta}_{MLE}$. The method is based on running R independent chains of Robbins-Monro updates in parallel, updating the quantities $\Delta_t^{(1)}, \dots, \Delta_t^{(R)}$ at each iteration following the iterative scheme

$$\begin{aligned} \Delta_t^{(r)} &= \Delta_{t-1}^{(r)} + \eta_t \nabla \log p(Y_t^{(r)}; x_t^{(r)}, \Delta_{t-1}^{(r)} + \hat{\theta}_{MLE}); \\ (Y_t^{(r)}, x_t^{(r)}) &\stackrel{\text{iid}}{\sim} \mathbb{F}_n \text{ for } r = 1, \dots, R \text{ and } t = 1, \dots, n \end{aligned} \quad (4)$$

where the learning rate is given by $\eta_t = \eta_0 t^{-c}$, $1/2 < c < 1$, $\eta_0 > 0$ and the starting point is set at $\Delta_0^{(r)} = 0$ for $r = 1, \dots, R$. Let $\bar{\Delta}_n^{(r)} = n^{-1} \sum_{s=1}^n \Delta_s^{(r)}$ be the average stochastic error at the end of the

procedure on the r -th chain. We evaluate the variability of such a stochastic quantity across the R chains via

$$\hat{V}_{n,R} = \frac{1}{R-1} \sum_{r=1}^R \left\{ \bar{\Delta}_n^{(r)} - \bar{\Delta}_{n,R} \right\} \left\{ \bar{\Delta}_n^{(r)} - \bar{\Delta}_{n,R} \right\}^T, \quad (5)$$

where $\bar{\Delta}_{n,R} = R^{-1} \sum_{r=1}^R \bar{\Delta}_n^{(r)}$ is the average stochastic error across chains. The update in Equation (4) is equivalent to a Robbins-Monro update centered around the MLE. Note, in fact, that by adding $\hat{\theta}_{MLE}$ on both sides of the update in Equation (4), one exactly retrieves the update (3). The choice of $\Delta_0^{(r)} = 0$, thus, implies the initialisation of (3) at $\hat{\theta}_{MLE}$. Such an initialisation makes sense only once the MLE has been obtained, and is not available when using stochastic approximations to obtain point estimates, as usually done in the literature. Nevertheless, even if started at the MLE, the variability of the stochastic gradients leads the Robbins-Monro updates to jiggle around the $\hat{\theta}_{MLE}$ rather than stopping there. ARM-B takes advantage of this intrinsic variability of the updates to estimate the asymptotic variance of $\hat{\theta}_{MLE}$.

Proposition 1. Under Assumptions 2 and 3, it holds that $\hat{V}_{n,R} - V(\mathbf{Q}_n(\mathbf{F})) \xrightarrow{P} 0$.

The proof of Proposition 1, which is reported in the Appendix, relies on the averaged construction of the quantities $\bar{\Delta}_n^{(1)}, \dots, \bar{\Delta}_n^{(R)}$ which allows to directly apply Theorem 2 in Reference [8]. At the same time, the proposition can be easily adapted to more general settings by taking advantage of the available extensions of the results in Reference [8] to more flexible assumptions. See, e.g., [7] for an extension to globally convex locally strongly convex log-likelihoods, or [5] for a functional alternative.

From an implementation perspective, ARM-B requires specifying the number of resamples R , the initial step η_0 , and the decay rate c . However, it is convenient to set c arbitrarily close to $1/2$ to slow down the decay and allow for larger updates; thus, we set $c = 1/2 + \epsilon$ and $\epsilon = 10^{-3}$. In addition, we equip the chains of updates with a burn-in period of length B , which aims to start the trajectory averaging after the chains have stabilised. The number of iterations to burn-in can be set in advance or evaluated as the algorithm iterates. For the latter option, we suggest monitoring the log-likelihood of the model and start the averaging procedure when its absolute percentage change drops under a given tolerance level, i.e., when $|(\ell_t - \ell_{t-1})|/|\ell_{t-1}| \leq \tau$, with $\ell_t = \log p(Y, X, \Delta_t + \hat{\theta})$ and $\tau > 0$. The pseudo-code for a single chain run is described in Algorithm 1. As concerns the initial learning rate η_0 , a careless specification may be detrimental to the performance of the algorithm. Therefore, we suggest tuning it by evaluating a single chain with no burn-in on a grid of candidate rates. The best-performing rate is then chosen by evaluating the log-likelihood of the model at the end of the chain, as reported in Algorithm 2. After this step, the R chains are then run using the selected learning rate. The full procedure to run the ARM-B is summarised in Algorithm 3.

4 | Simulation Experiments

This section investigates the performance of the ARM-B estimator in quantifying the asymptotic variability of the MLE,

ALGORITHM 1 | ARM chain.

Input: $\hat{\theta}, B_{\max}, \eta_0$
Data: Y, X, n

- 1 $\Delta_0 \leftarrow 0; \bar{\Delta}_0 \leftarrow 0; B \leftarrow B_{\max};$
- 2 **for** t **from** 1 **to** $B + n - 1$ **do**
- 3 $\eta_t \leftarrow \eta_0 t^{-.5001};$
- 4 $(y_t, x_t) \sim \mathbb{F}_n;$
- 5 $\Delta_t \leftarrow \Delta_{t-1} + \eta_t \nabla \log p(y_t; x_t, \Delta_{t-1} + \hat{\theta});$
- 6 **if** $t \leq B$ **then**
- 7 $\ell_t \leftarrow \log p(Y, X, \Delta_t + \hat{\theta});$
- 8 **if** ℓ_t *converged* **then**
- 9 $B \leftarrow t;$
- 10 **else**
- 11 $\bar{\Delta}_{t-B} \leftarrow \{(t - B - 1)\bar{\Delta}_{t-1} + \Delta_t\} / (t - B);$

Return: $\bar{\Delta}_n;$

ALGORITHM 2 | Learning rate selection.

Input: $\hat{\theta}, \eta_0$
Data: Y, X, n

- 1 **for** s **from** 1 **to** 10 **do**
- 2 **Evaluate** $\bar{\Delta}_n^{(s)}$ *via Algorithm 1*, with arguments
 $(\hat{\theta}, 0, \eta_0 / 2^{s-1});$
 Store: $\bar{\Delta}_n^{(s)};$
- 3 $s^* \leftarrow \operatorname{argmax}_s \log p(Y, X, \Delta_n^{(s)} + \hat{\theta});$
- 4 $\eta_0^* \leftarrow \eta_0 / 2^{s^*-1};$

Return: $\eta_0^*;$

ALGORITHM 3 | ARM-B.

Input: $\hat{\theta}, R, B_{\max}, \eta_0$
Data: Y, X, n

- 1 **Evaluate** η_0^* *via Algorithm 2* with arguments $(\hat{\theta}, \eta_0);$
- 2 **for** r **from** 1 **to** R **do**
- 3 **Evaluate** $\bar{\Delta}_n^{(r)}$ *via Algorithm 1*, with arguments $(\hat{\theta}, B_{\max}, \eta_0^*);$
 Store: $\bar{\Delta}_n^{(r)};$
- 4 **Compute** $\hat{V}_{n,R}$ *from* $\bar{\Delta}_n^{(1)}, \dots, \bar{\Delta}_n^{(R)}$ *via* (5);

Return: $\hat{V}_{n,R};$

considering the case of logistic and Poisson regression models. Simulations run under different setting dimensions and specifications of the generative distribution of the design matrix. To benchmark the accuracy of the estimator, we use a parametric bootstrap as an oracle method, with $R = 500$ datasets simulated from the true model. The same oracle has been used in Reference [12] with a higher number of replications on settings involving fewer parameters. We then evaluate the mean square distance of the diagonal of the estimated covariance matrix from the oracle one. For assessing ARM-B, we compare its performance with that of the nonparametric Bootstrap and BLB.

As mentioned, we run the experiments for different configurations of the design matrix. In particular, following the simulation setups of [3], we generate the covariates according to $x_i \sim \mathcal{N}(0, \Sigma)$ for $i = 1, \dots, n$, with Σ specified as

- Identity: $\Sigma = I_p;$
- Toeplitz: $\Sigma_{i,j} = 0.5^{|i-j|};$
- Equicorrelation: $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = 0.2$ for $i \neq j.$

The unique combinations of such configurations with $n \in \{5,000, 10,000, 50,000, 100,000\}$ and $p \in \{50, 100, 200, 500\}$ define the grid of settings analysed. Therefore, the ratio p/n goes from 5×10^{-4} to 10^{-1} . In the [Supporting Information](#) are included additional simulation experiments with $p/n \in \{0.5, 0.2, 0.1\}.$

As concerns the software, all analyses are conducted in R [14]. The Bootstrap methods rely on the resampling functions provided by the `boot` package [15], while the BLB resampling mechanism employs the implementation of the `rSW2utils` package [16]. Note that `rSW2utils` adopts adaptive convergence checking for BLB. If convergence is reached, the algorithm stops before running all the Monte Carlo iterations and, thus, speeds up the estimation. On each Bootstrap and BLB resample, the model is fitted via the standard `glm` function, which runs on C code. The stochastic updates of Algorithm 3 instead run via a custom implementation using Rcpp [17].

4.1 | **Logistic Regression**

Following [12], the true parameter values are all set to $1/\sqrt{p}$ across the different settings, and data are generated from a logistic regression model. We run ARM-B, Bootstrap, and BLB with $R = 100$ Monte Carlo iterations. For ARM-B, we run Algorithm 3 with $B_{\max} = n$ and $\eta_0 = 1$. The tolerance level to end the burn-in period is set at 10^{-2} and evaluates the percentage change of the objective function after periods of $n/5$ iterations. Recall that η_0 is only a plausible candidate learning rate. A well-performing value is then automatically chosen by Algorithm 3. Additional experiments with fixed stepsize values and no tuning are reported in the [Supporting Information](#). For what concerns BLB, [12] suggest using $\gamma = 0.7$ and $s \in \{1, 2, 3\}$, where s is the number of sampled subsets and n^γ their size. Thus, we fix $\gamma = 0.7$ and let $s = 1$ since we do not observe significant differences for higher values of s , apart from increased computational times. Additional simulations with higher values of s , γ , and R can be found in the [Supporting Information](#), as well as experiments based on lower sample sizes.

Figure 1 presents the performances of the different methods in terms of mean square distance of the diagonal of the estimated covariance matrix from the oracle. Note that a hard-coded upper visual limit has been set on the vertical axis at 0.1 because the performance of BLB with $n = 5,000$ sometimes appears out of scale compared to the other two methods and does not allow a clear visual investigation of the results. As expected, with p fixed, the accuracy of all methods improves as the sample size increases. The only exception is the equicorrelation setting, where BLB appears to suffer most convergence issues. However, ARM-B exhibits a very different pattern from that of BLB. It is apparent, in fact, that BLB slightly outperforms ARM-B when the ratio p/n is particularly favourable. Such settings are also more in line with the experiments in Reference [12]. However, when p/n increases, BLB estimates appear much more unstable. The poor

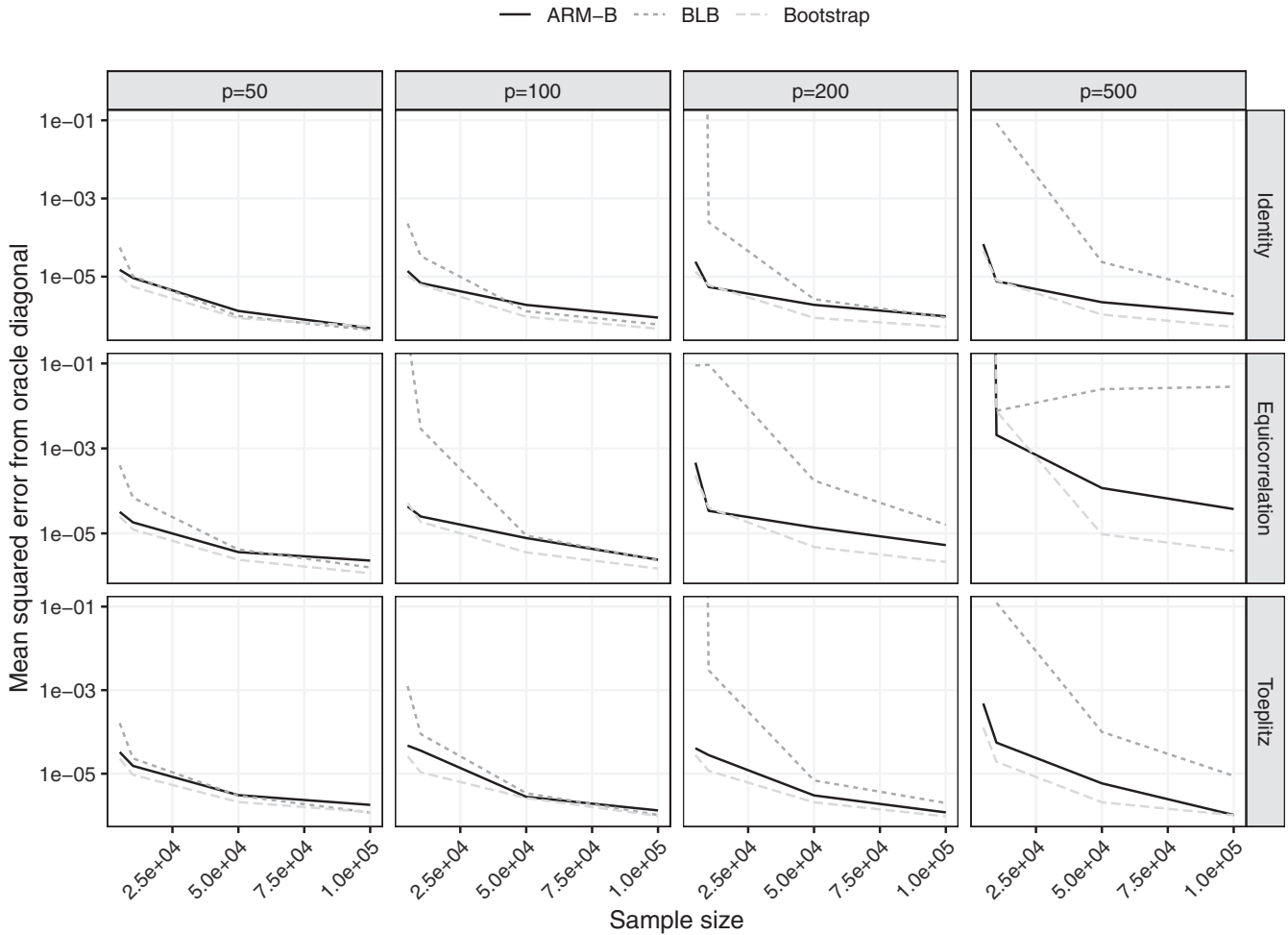


FIGURE 1 | Simulation results for the logistic regression model in terms of mean square distance of the diagonal of the estimated covariance matrix from the oracle.

performance of the BLB configuration in the settings with higher p/n can be explained by the fact that each BLB resample contains at most n^γ data points, and thus $\gamma = 0.7$ might not be enough when the number of parameters increases. The [Supporting Information](#) shows how BLB performances improve when increasing γ but partially sacrifice their computational convenience. However, ARM-B appears more robust to increases in p/n and to different configurations of Σ . In the settings with $p = 200$ or $p = 500$, it is almost always preferable to BLB and needs a lower n to get accurate performances. As reported in Figure 2, computational times endorse the same flipping preference between BLB and ARM-B based on the ratio p/n . When p/n is low, BLB slightly outperforms ARM-B both statistically and computationally. In more challenging settings, when p/n increases, ARM-B remains statistically reliable and still computationally viable, in contrast to BLB.

4.2 | Poisson Regression

Coherently with the logistic simulation settings, the true parameters are set to $1/\sqrt{p}$ in all experiments related to the Poisson regression case. As before, we set $R = 100$ for all methods and keep the same configuration of ARM-B. Concerning BLB, we set $\gamma = 0.7$ and $s = 2$ because its influence on the accuracy

performance was more evident than in the logistic regression case. However, we experienced systematic numerical instability when fitting the models with Σ following an equicorrelation or Toeplitz design due to the generation of extremely high counts in the simulated data. Thus, in this case, we only present the results of the case of $\Sigma = I_p$. Figure 3 outlines the accuracy performance of the three methods and their computational times. Different from the logistic regression experiments, the performance of BLB appears to be more robust across the different settings. The flat lines related to computational times point out that, for increasing sample sizes, BLB reaches convergence faster and stops before using all Monte Carlo iterations. Nevertheless, similarly to the logistic regression case, the performance of BLB deteriorates with p/n increasing. In such cases, ARM-B appears again as a more robust alternative, as in Section 4.1. The [Supporting Information](#) reports additional experiments for different values of s , γ , and R .

5 | Application

We use the ARM-B estimator to quantify the uncertainty around logistic regression parameters estimated on the Air Pressure System (APS) Failure and Operational Data from Scania. The dataset has been shared at the 15th International Symposium on Intelligent Data Analysis. It collects operational data from heavy Scania

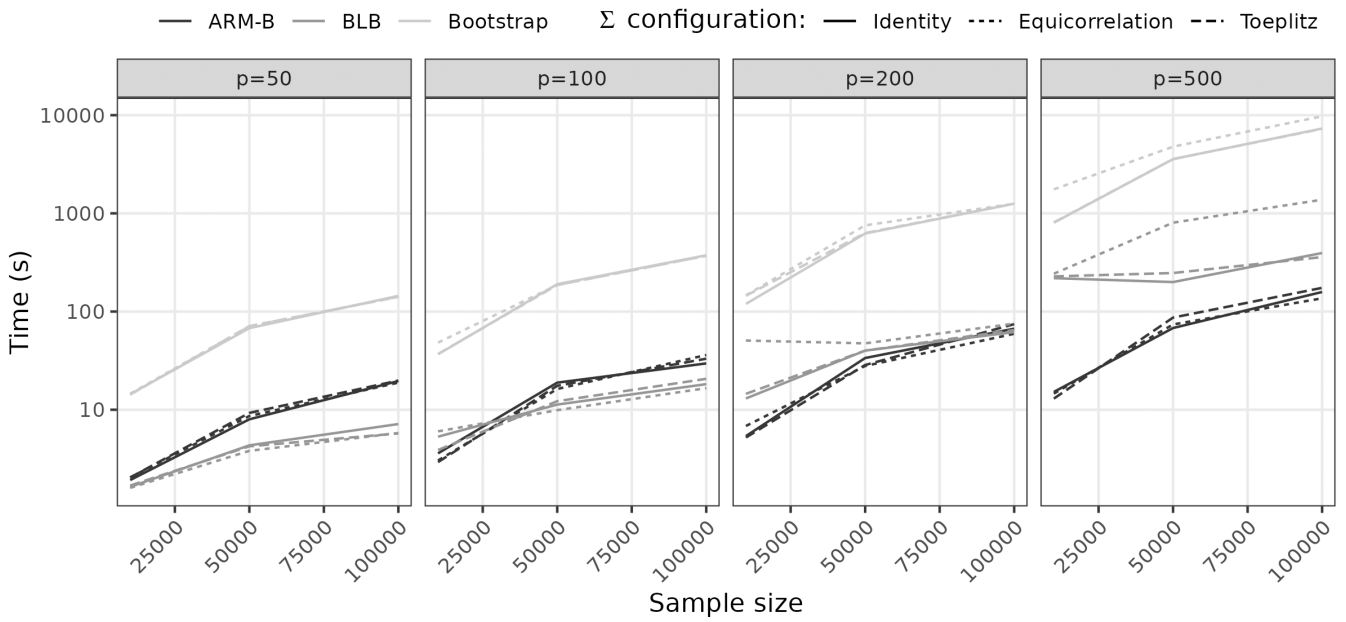


FIGURE 2 | Computational times from the simulations experiments on the logistic regression model.

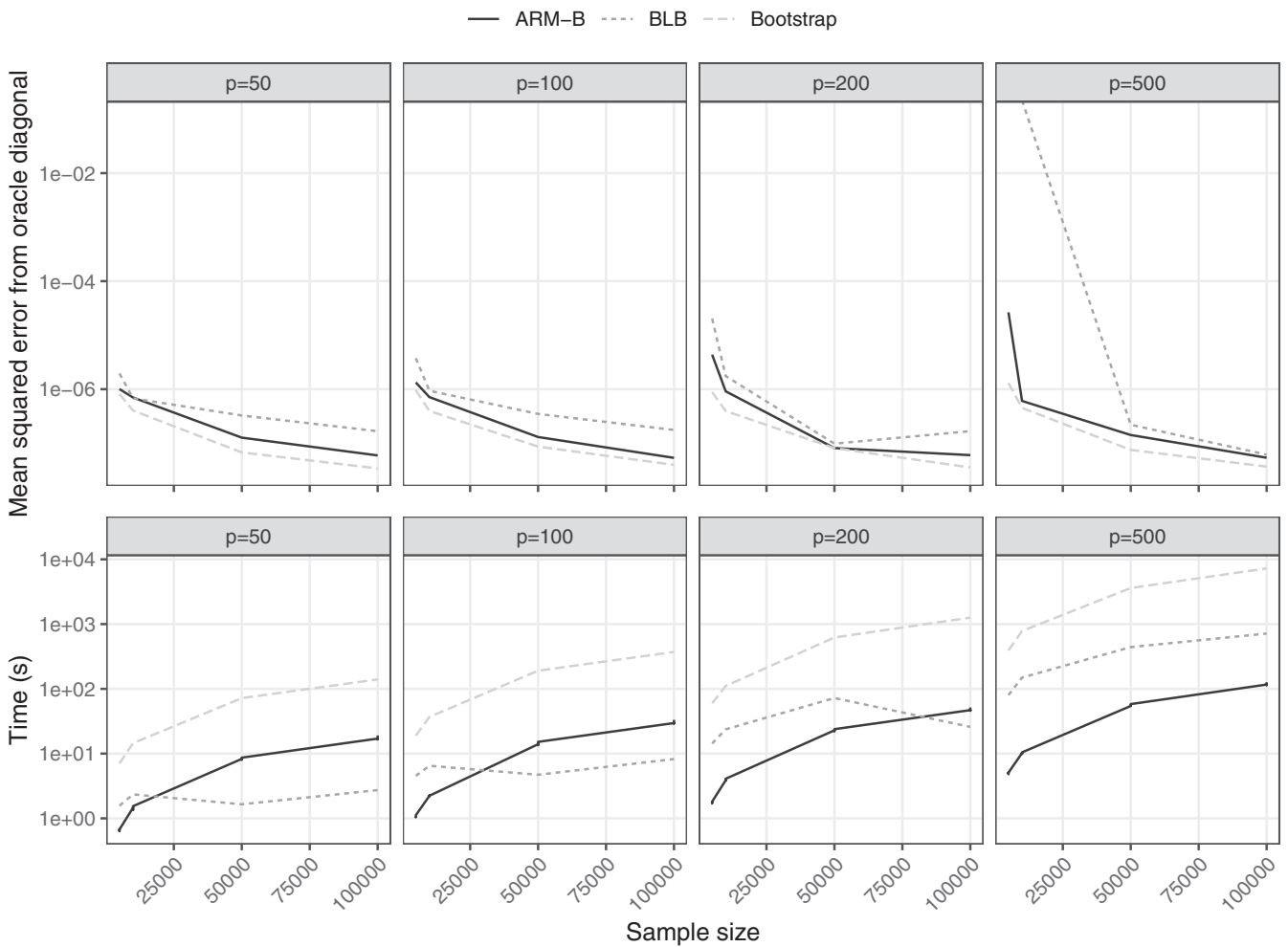


FIGURE 3 | Simulation results for the Poisson regression model in terms of mean square distance of the diagonal of the estimated covariance matrix from the oracle and computational times.

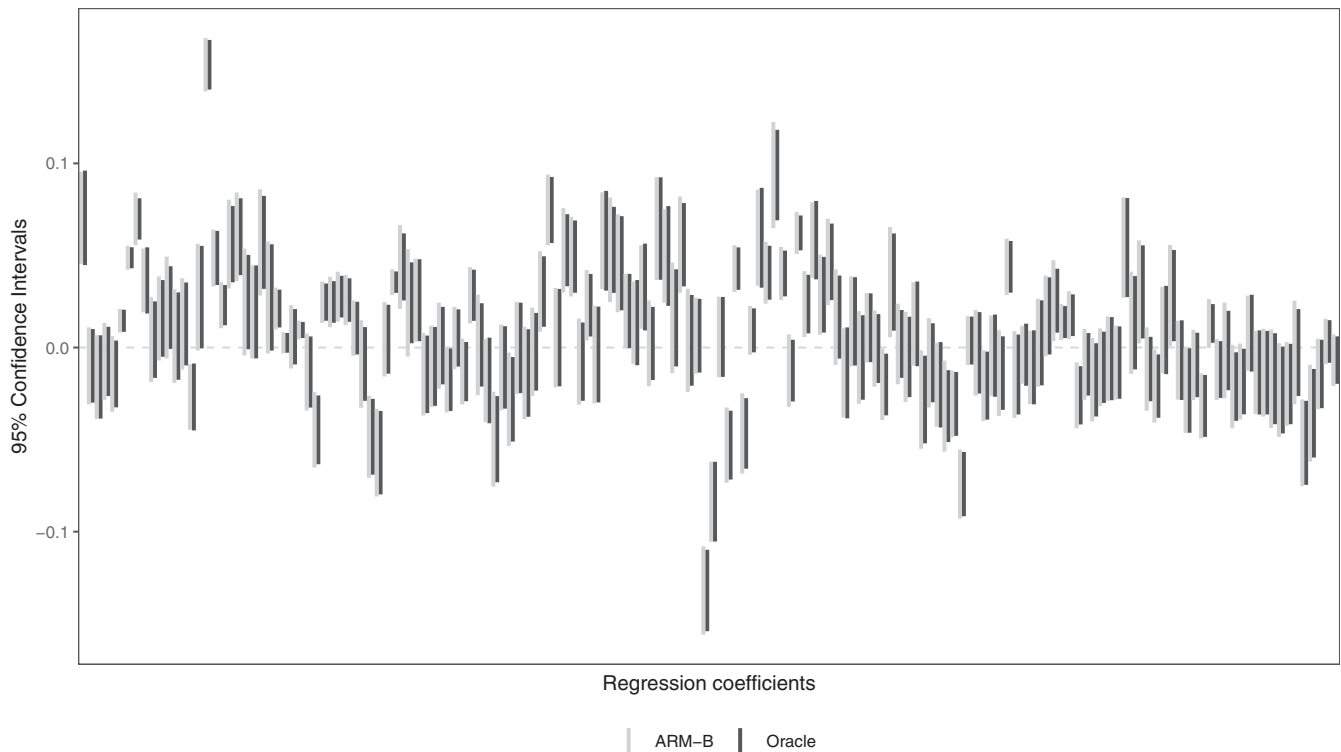


FIGURE 4 | Comparison of ARM-B asymptotic confidence intervals with the oracle ones on the APS failure dataset.

trucks, along with a binary response variable denoting the failure or proper functioning of the Air pressure system of the trucks, for a total of 60,000 observations on 170 numerical variables, outcome excluded. Note that covariate names have been anonymised for proprietary reasons. Thus, we are not able to interpret parameter values directly. While the dataset has been analysed for predictive aims [18], we argue that inference about parameter estimates might be of crucial importance from a business point of view, allowing the identification of the variables playing a statistically significant role in the failure of the APS.

From a pre-processing perspective, the dataset presents two main challenges. The first is the presence of missing values ($\approx 8\%$), and the second is the severe imbalance in the outcome class, with only 1000 failures out of the 60,000 observations. To deal with the first issue, we retain only rows with less than 30% missing rate, and from the subsetted dataset, we drop all columns with more than 70% missing rate, ending up with $n = 57143$ and $p = 163$. Finally, we replace the remaining missing values ($\approx 3.5\%$) by simply imputing the median by column. Regarding the balancing issue, we rely on the smoothed bootstrap approach proposed in Reference [19], as implemented in the R package ROSE [20]. That is, first, we compute the MLE on a balanced resample from ROSE. Second, we let Algorithm 3 sample new data points at each iteration via ROSE rather than directly using the empirical distribution of the dataset. ARM-B is run with $B = n$ and $\eta_0 = 1$ as in the simulation experiments. To benchmark the accuracy of ARM-B, we compute the oracle estimates with $R = 500$ as in the simulation experiments. However, differently from the simulation studies, the true parameters are not available in this case. Therefore, we leveraged the bootstrap functionalities made available by the ROSE package, and we employed as an oracle the

covariance matrix estimated from the logistic regressions fitted on the R smoothed bootstrap resamples provided by the package. Notice that the ratio p/n might allow BLB to perform well, yet adapting its implementation to deal with imbalanced data is not obvious. Consequently, we only compare ARM-B to the oracle estimates described above.

Figure 4 shows that the ARM-B estimates of the asymptotic standard deviations closely approximate the oracle ones and can be used in constructing the confidence intervals to draw inferences on MLE parameters. The average length ratio between the oracle and ARM-B confidence intervals is 0.985, which points out a close accordance between the two methods. Running the estimation with 4 cpus on a personal laptop,² ARM-B only took around 140 s, against almost 24 min needed by the oracle method.

6 | Discussion

In this paper, we presented the ARM-B estimator, a computationally scalable procedure aimed at assessing the variability of the MLE on massive datasets. From a practical perspective, ARM-B starts from a single pointwise evaluation of the MLE, which practitioners can obtain with their numerical method of choice. In this regard, the convenience of ARM-B is tangible in all those settings where practical time constraints allow evaluating the MLE once but prevent using traditional bootstrap methods to draw statistical inference on model parameters.

The simulation experiments stress the robustness of the ARM-B estimator to different settings and data dimensions, highlighting, in particular, its statistical and computational convenience

compared to competitors when the number of parameters increases. While the simulations in Section 4 focus on logistic and Poisson regression, in the [Supporting Information](#), we outline a possible use of ARM-B on random effects models. Nevertheless, unlike traditional bootstrap methods, which allow practitioners to rely on existing software to compute the MLE for the model of interest, it is worth emphasising that ARM-B requires directly providing a function that evaluates the log-likelihood of the model and its gradient.

From a theoretical point of view, the consistency of ARM-B can be easily extended to more complex models than the ones considered in this paper, following recent extensions of standard asymptotic results for averaged stochastic estimators (e.g., [7], [5]).

Computationally, the method still requires running multiple independent chains of stochastic updates, eventually in parallel, mimicking the behaviour of standard nonparametric bootstrap procedures. However, if the computational burden needs to be lowered further, the sample distribution of the independent chains can be approximated by relying on techniques proposed for online inference in the stochastic optimisation literature (e.g., [4], [3]).

Reproducibility

The code to reproduce the simulations and real data results presented in the paper and the [Supporting Information](#) is available at the online repository https://github.com/giuseppealfonzetti/armb_experiments.

Acknowledgments

This work was supported by the Departmental Strategic Plan (PSD) of the University of Udine, Interdepartmental Project on Artificial Intelligence (2022–2025). A preliminary version of this work was presented at the Statistics and Data Science 2024 Conference (April 2024, Palermo, Italy). Open access publishing facilitated by Università degli Studi di Udine, as part of the Wiley - CRUI-CARE agreement.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The APS Failure dataset is publicly available in the UCI Machine Learning Repository at <https://archive.ics.uci.edu/dataset/421/aps+failure+at+scania+trucks>.

Endnotes

¹ <https://archive.ics.uci.edu/dataset/421/aps+failure+at+scania+trucks>.

² Specifications: AMD Ryzen 7 4800H 16 × 2.9 GHz, R 4.4.1, gcc 11.4.0, Ubuntu 22.04.

References

1. H. Robbins and S. Monro, “A Stochastic Approximation Method,” *Annals of Mathematical Statistics* 22, no. 3 (1951): 400–407.

2. L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization Methods for Large-Scale Machine Learning,” arXiv:1606.04838, (2018).

3. X. Chen, J. D. Lee, X. T. Tong, and Y. Zhang, “Statistical Inference for Model Parameters in Stochastic Gradient Descent,” *Annals of Statistics* 48, no. 1 (2020): 251–273.

4. Y. Fang, J. Xu, and L. Yang, “Online Bootstrap Confidence Intervals for the Stochastic Gradient Descent Estimator,” *Journal of Machine Learning Research* 19, no. 1 (2018): 3053–3073.

5. S. Lee, Y. Liao, M. H. Seo, and Y. Shin, “Fast and Robust Online Inference With Stochastic Gradient Descent via Random Scaling,” *Proceedings of the AAAI Conference on Artificial Intelligence* 36, no. 7 (2022): 7381–7389.

6. P. Toulis and E. M. Airoldi, “Asymptotic and Finite-Sample Properties of Estimators Based on Stochastic Gradients,” *Annals of Statistics* 45, no. 4 (2017): 1694–1727.

7. W. J. Su and Y. Zhu, “HiGrad: Uncertainty Quantification for Online Learning and Stochastic Approximation,” *Journal of Machine Learning Research* 24, no. 124 (2023): 1–53.

8. B. T. Polyak and A. B. Juditsky, “Acceleration of Stochastic Approximation by Averaging,” *SIAM Journal on Control and Optimization* 30, no. 4 (1992): 838–855.

9. A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application* (Cambridge University Press, 1997).

10. B. Efron and T. Hastie, *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science* (Cambridge University Press, 2021).

11. B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman and Hall/CRC, 1994).

12. A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, “A Scalable Bootstrap for Massive Data,” *Journal of the Royal Statistical Society, Series B: Statistical Methodology* 76, no. 4 (2014): 795–816.

13. H. White, “Maximum Likelihood Estimation of Misspecified Models,” *Econometrica* 50, no. 1 (1982): 1–25.

14. R Core Team, *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, 2025).

15. A. Canty and B. D. Ripley, “Boot: Bootstrap R (S-Plus) Functions,” *R Package Version 1* (2025): 3–31.

16. D. Schlaepfer, “rSW2utils: Utility Tools for SOILWAT2 and STEP-WAT2 Simulation Experiments,” *R Package Version 0*, no. 2 (2025): 1.

17. D. Eddelbuettel and R. François, “Rcpp: Seamless R and C++ Integration,” *Journal of Statistical Software* 40, no. 8 (2011): 1–18.

18. C. F. Costa and M. A. Nascimento, “Ida 2016 Industrial Challenge: Using Machine Learning for Predicting Failures,” in *Advances in Intelligent Data Analysis XV*, ed. H. Boström, A. Knobbe, C. Soares, and P. Papapetrou (Springer International Publishing, 2016), 381–386.

19. G. Menardi and N. Torelli, “Training and Assessing Classification Rules With Imbalanced Data,” *Data Mining and Knowledge Discovery* 28, no. 1 (2014): 92–122.

20. N. Lunardon, G. Menardi, and N. Torelli, “Rose: A Package for Binary Imbalanced Learning,” *R Journal* 6 (2014): 79–89.

21. A. W. van der Vaart, *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics (Cambridge University Press, 1998).

Supporting Information

Additional supporting information can be found online in the Supporting Information section. Supplementary Material Data S1: Supporting Information.

Appendix A

Proof of Proposition 1

Let $\mathcal{F}_t^{(r)}$ be the filtration on the r -th chain at iteration t , and denote the collection of such filtrations with $\mathcal{F}_t^R = \{\mathcal{F}_t^{(1)}, \dots, \mathcal{F}_t^{(R)}\}$. Under Assumptions 1–3 we can directly apply Theorem 2 in Reference [8] to show that the quantity $\bar{\Delta}_t = t^{-1} \sum_{s=1}^t \Delta_s$ follows

$$\sqrt{t} \bar{\Delta}_t = t^{-1/2} \sum_{s=1}^t \hat{H}_n^{-1} \nabla \log p(Y_s; x_s, \hat{\theta}_{MLE}) + o_p(1),$$

which immediately leads to $t \text{Var}_{\mathbb{F}_n} \{\bar{\Delta}_t | \mathcal{F}_t\} \xrightarrow{p} \hat{H}_n^{-1} \hat{J}_n \hat{H}_n^{-1}$, where $\hat{H}_n = -\text{E}_{\mathbb{F}_n} \{\nabla^2 \log p(Y; x, \hat{\theta}_{MLE})\}$ and $\hat{J}_n = \text{Var}_{\mathbb{F}_n} \{\nabla \log p(Y; x, \hat{\theta}_{MLE})\}$.

After $t = n$ iterations, the observations $\bar{\Delta}_n^{(1)}, \dots, \bar{\Delta}_n^{(R)}$ in Algorithm 3 are independent and identically distributed conditioned on \mathcal{F}_n^R , given the sampling procedure in Equation (4). It follows that the empirical variance estimator provided by $\hat{V}_{n,R}$ is consistent for the conditional variance of $\bar{\Delta}_n$ given the filtration set \mathcal{F}_n^R , i.e., $\hat{V}_{n,R} \xrightarrow{p} \text{Var}_{\mathbb{F}_n} \{\bar{\Delta}_n | \mathcal{F}_n^R\}$.

By combining the two results, we obtain therefore that $\hat{V}_{n,R} - \hat{H}_n^{-1} \hat{J}_n \hat{H}_n^{-1} \xrightarrow{p} 0$, which, by the continuous mapping theorem [21], implies that $\hat{V}_{n,R} - V(\mathbf{Q}_n(\mathbf{F})) \xrightarrow{p} 0$, given that $\hat{\theta}_{MLE} \xrightarrow{a.s.} \theta^*$ [13].