SCIENCE CHINA
Mathematics

• ARTICLES •

September 2025 Vol. 68 No. 9: 2207–2230
https://doi.org/10.1007/s11425-024-2350-5

Energy distance-based subsampling Markov chain Monte Carlo

Sumin Wang[1], Fasheng Sun[2] & Min-Qian Liu[3,*]

[1]*NITFID, LPMC & KLMDASR, Center for Combinatorics, Nankai University, Tianjin* 300071*, China;*
[2]*School of Mathematics and Statistics & KLAS, Northeast Normal University, Changchun* 130024*, China;*
[3]*NITFID, LPMC & KLMDASR, School of Statistics and Data Science, Nankai University, Tianjin* 300071*, China*

*Email: wangsm*088*@nankai.edu.cn, sunfs*359*@nenu.edu.cn, mqliu@nankai.edu.cn*

Received February 2, 2024; accepted October 11, 2024; published online March 27, 2025

Abstract Subsampling plays a crucial role in enhancing the efficiency of Markov chain Monte Carlo (MCMC) algorithms. This paper presents a subsampling-based MCMC algorithm aimed at addressing the computational complexity challenges of traditional MCMC methods on large-scale datasets. The proposed approach significantly reduces computational costs by approximating the full data likelihood function using only a subset of the full data in each iteration. The subsampling process is guided by the fidelity to the full data, which is measured by the energy distance. The resulting algorithm, termed the energy distance-based subsampling MCMC (EDSS-MCMC), offers a flexible approach while maintaining the simplicity of the standard MCMC algorithm. Additionally, we provide an analysis of the invariant distribution generated by the EDSS-MCMC algorithm and quantify the total variation norm between this distribution and the target distribution. Numerical experiments demonstrate the outstanding performance of the proposed algorithm on large-scale datasets. Compared with the standard MCMC algorithm and other subsampling MCMC algorithms, the EDSS-MCMC algorithm exhibits advantages in terms of accuracy and computational speed. Therefore, the proposed algorithm holds practical significance in tasks involving large-scale dataset analysis and machine learning.

Keywords Bayesian inference, energy distance, large-scale dataset, Markov chain Monte Carlo, subsampling

MSC(2020) 62D05, 62F15

Citation: Wang S M, Sun F S, Liu M-Q. Energy distance-based subsampling Markov chain Monte Carlo. Sci China Math, 2025, 68: 2207–2230, https://doi.org/10.1007/s11425-024-2350-5

1 Introduction

Bayesian methods offer a flexible mathematical framework to estimate parameters and their uncertainty. In general, the posterior distribution cannot be analytically integrated, necessitating the use of approximations. Markov chain Monte Carlo (MCMC) [7] methods are commonly employed for Bayesian inference as they approximate the posterior distribution through a set of samples obtained from a Markov chain with the posterior distribution as its invariant distribution. Paradoxically, the standard MCMC algorithms are not scalable to large-scale datasets due to their computationally expensive nature.

[*] Corresponding author

Consequently, MCMC methods are not feasible for efficient runtime. This issue has sparked extensive research in recent years [5].

In recent years, many efforts have been made to develop scalable MCMC algorithms, including those methods based on approximations of the Metropolis-Hastings (MH) [10, 17] algorithms. Two main categories of approaches have emerged: divide-and-conquer methods and subsampling techniques. A divide-and-conquer method involves dividing the full data into batches, running MCMC on each batch separately, and then combining the results to obtain an approximation of the posterior distribution [21,23]. On the other hand, a subsampling technique aims to reduce the number of samples at each iteration of the MH algorithm. Under the framework of subsampling techniques, the pseudo-marginal MH algorithm [2,3] is an exact subsampling approach, which relies on positive unbiased estimators (based on a subset of samples) of an unnormalized version of the target distribution. However, this approach remains (for now) mostly theoretical because such estimators are in general not available [12]. Attempts to circumvent the positivity and unbiasedness requirements of the estimator have been studied in [22]. In both cases, the authors resorted to sophisticated control variates, which can be computationally expensive to compute. Alternatively, approximate subsampling approaches have been proposed. These methods approximate the full data log-likelihood using subsets of samples obtained through uniform or nonuniform subsampling probabilities. However, these subsamples may not adequately cover the full data. For example, Figure 1 illustrates 100 subsamples obtained from a dataset of 1,000 samples using uniform [5] (see Figure 1(a)) and nonuniform [11] (see Figure 1(b)) subsampling probabilities. It is evident that the subsamples obtained through these two methods do not evenly cover the entire dataset and exhibit poor representativeness. This can be explained by the fact that although the subsampling methods are different, both methods involve random subsampling, which can lead to undersampled areas or clustering due to randomness.

In this paper, we propose a novel subsampling MCMC algorithm. The main idea is to use energy distance to reduce the full data into a set of optimal subsamples. This approach aims to achieve better coverage of the full data and improve the representativeness of the subsamples. We term the resulting algorithm as the energy distance-based subsampling MCMC (EDSS-MCMC). Compared with existing uniform subsampling MCMC algorithms [4,13] and nonuniform subsampling MCMC algorithms [11], the proposed algorithm yields a more accurate approximation with the same subsample size.

The rest of this paper is organized as follows. Section 2 provides background information regarding the problem of interest, the standard MH algorithm, and two classes of random subsampling MCMC algorithms. Section 3 outlines a deterministic approach to reduce the full data into an optimal set of subsamples and presents the corresponding EDSS-MCMC algorithm. In this section, we also examine the

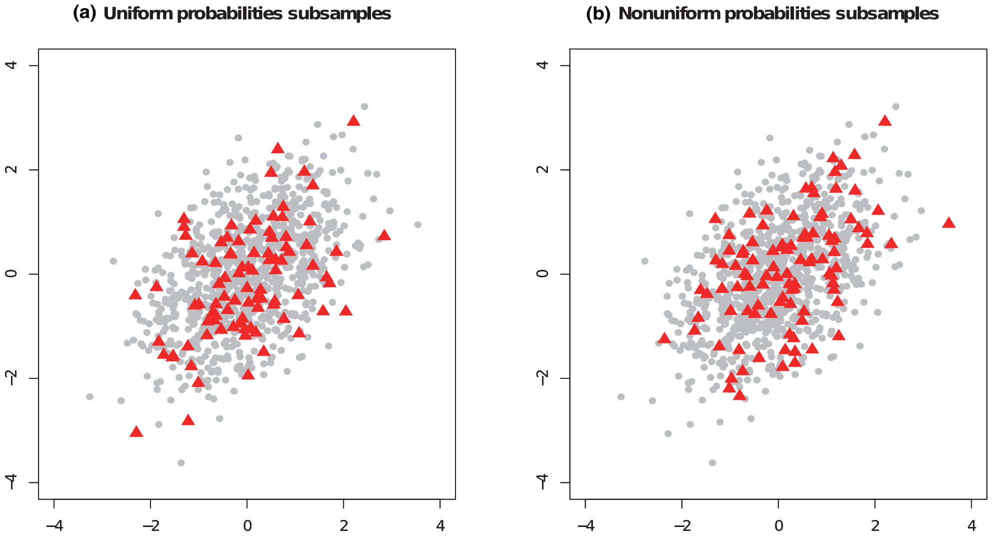


Figure 1 (Color online) 100 subsamples from 1,000 full data, where the subsamples are denoted by triangles

transition kernel of the EDSS-MCMC algorithm and demonstrate its ability to generate a Markov chain that approximates the target distribution. Additionally, we quantify the upper bound of the total variation norm between this invariant distribution and the target distribution. Section 4 conducts a series of numerical experiments to assess the effectiveness of the EDSS-MCMC, comparing it with standard (non-subsampling) MCMC techniques as well as other subsampling MCMCs. Section 5 demonstrates the proposed method using two real datasets. Section 6 concludes with a summary of the paper. All proofs are deferred to Appendix A.

2 Background and related work

Suppose that

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \overset{\text{i.i.d.}}{\sim} F \ (\boldsymbol{x} \in \mathcal{X}, \mathcal{X} = [0,1]^p),$$

where the underlying density f of $\boldsymbol{x}$ is fully specified by a d-dimensional parameter vector $\theta \in \Theta$ ($\Theta \in \mathbb{R}^d, d > 0$). With $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the associated likelihood can be denoted by

$$f(X|\theta) = \prod_{i=1}^{N} f(\boldsymbol{x}_i|\theta).$$

In the Bayesian approach, θ is assumed to be a random vector with a prior distribution $p(\theta)$. The posterior distribution π is defined on the measurable space $(\Theta, \mathcal{B}(\Theta))$ and its density $\pi(\cdot|X)$ satisfies

$$\pi(\theta|X) = \frac{\prod_{i=1}^{N} f(\boldsymbol{x}_i|\theta)p(\theta)}{\int_\Theta \prod_{i=1}^{N} f(\boldsymbol{x}_i|\theta)p(\theta)d\theta} \propto \prod_{i=1}^{N} f(\boldsymbol{x}_i|\theta)p(\theta) \quad \text{for any } \theta \in \Theta. \tag{2.1}$$

Bayesian inference relies on the posterior distribution π. However, in many cases, π has a complex form that is analytically infeasible to find. To address this issue, the MH algorithm is often employed to generate samples from the posterior distribution for statistical inference.

2.1 The standard MH algorithm

A standard approach to sample approximately from π is the MH algorithm. This algorithm involves constructing an ergodic Markov chain with the invariant distribution π. Given a conditional proposal distribution $q(\cdot|\theta)$, and assume that the MH Markov chain is at state θ_k, a transition $\theta_k \to \theta_{k+1}$ consists of the following two steps:

(i) proposing a new parameter $\theta' \sim q(\cdot|\theta_k)$;

(ii) setting the next state of the Markov chain as $\theta_{k+1} = \theta'$ with probability

$$\alpha(\theta_k, \theta') = 1 \wedge a(\theta_k, \theta'), \quad a(\theta_k, \theta') = \frac{\pi(\theta'|X)q(\theta_k|\theta')}{\pi(\theta_k|X)q(\theta'|\theta_k)}, \tag{2.2}$$

where $a \wedge b$ denotes the minimum of a and b.

In practice, the accept/reject step of the MH step is implemented by sampling a uniform random variable $u \sim U(0,1)$ and accepting θ' if and only if

$$u < a(\theta_k, \theta') = \frac{\pi(\theta'|X)q(\theta_k|\theta')}{\pi(\theta_k|X)q(\theta'|\theta_k)}.$$

If we define the average log-likelihood ratio as $\Lambda(\theta_k, \theta') = l(\theta') - l(\theta_k)$, where

$$l(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log f(\boldsymbol{x}_i|\theta)$$

represents the average of the full data log-likelihood, and define

$$\psi(u, \theta_k, \theta') = \frac{1}{N} \log \left[u \frac{p(\theta_k)q(\theta'|\theta_k)}{p(\theta')q(\theta_k|\theta')} \right],$$

then we accept θ' if and only if

$$\Lambda(\theta_k, \theta') > \psi(u, \theta_k, \theta'). \tag{2.3}$$

For completeness and ease of discussion, we present the standard MH algorithm in Algorithm 1 below.

Algorithm 1: The standard MH algorithm

```
Input: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, $q(\cdot|\theta)$
Output: $\theta_1, \ldots, \theta_T$
for $k = 1, \ldots, T$ do
    $\theta \leftarrow \theta_k$;
    $\theta' \sim q(\cdot|\theta)$, $u \sim U(0,1)$;
    $\psi(u, \theta, \theta') \leftarrow \frac{1}{N}\log[u\frac{p(\theta)q(\theta'|\theta)}{p(\theta')q(\theta|\theta')}]$;
    $\Lambda(\theta, \theta') \leftarrow \frac{1}{N}\sum_{i=1}^{N}\log[\frac{f(\boldsymbol{x}_i|\theta')}{f(\boldsymbol{x}_i|\theta)}]$;
    if $\Lambda(\theta, \theta') > \psi(u, \theta, \theta')$ then
        $\theta_{k+1} = \theta'$ {Accept};
    else
        $\theta_{k+1} = \theta$ {Reject};
    end
end
```

2.2 MCMC algorithms with subsampling log-likelihood

In the standard MH algorithm (Algorithm 1), evaluating $\Lambda(\theta, \theta')$ in each iteration requires computing the average of the full data log-likelihood, $l(\theta)$ and $l(\theta')$. This can be computationally demanding for large-scale datasets. To overcome this challenge, subsampling MCMC algorithms take advantage of subsampling to approximate $l(\theta)$. This approximation serves as the basic idea underlying this approach. Here, we introduce two classes of approximate subsampling MCMC algorithms.

Uniform subsampling MCMC (Uniform-MCMC). In each MH iteration, Bardenet et al. [4, 5] and Korattikara et al. [13] proposed using uniform subsamples to approximate $l(\theta)$ and $l(\theta')$. Specifically, let $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_n^*$ represent n random samples taken from the full data with replacement, following uniform subsampling probabilities. To approximate $l(\theta)$, Bardenet et al. [5] and Korattikara et al. [13] suggested using

$$l_n^{\mathrm{Uni}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log f(\boldsymbol{x}_i^*|\theta).$$

Thus, in each MH iteration, $\Lambda(\theta, \theta')$ can be replaced by

$$\Lambda_n^{\mathrm{Uni}}(\theta, \theta') = l_n^{\mathrm{Uni}}(\theta') - l_n^{\mathrm{Uni}}(\theta)$$

to speed up the MCMC process.

In addition to uniform subsampling, we now introduce the nonuniform subsampling approach, which provides a more accurate approximation of $l(\theta)$.

Most likely optimal subsampling MCMC (MLO-MCMC). To improve the approximation efficiency, Hu and Wang [11] focused on nonuniform subsamples. Let $\eta_1, \eta_2, \ldots, \eta_N$ denote the nonuniform subsampling probabilities such that $\sum_{i=1}^{N}\eta_i = 1$. Let $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_n^*$ be the subsamples randomly drawn with replacement according to η_i's. Consequently, $l(\theta)$ can be approximated by

$$l_n^*(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{N\eta_i}\log f(\boldsymbol{x}_i^*|\theta).$$

Since $l_n^*(\theta)$ is an unbiased estimator of $l(\theta)$, Hu and Wang [11] proposed to minimize the variance of $l_n^*(\theta)$ to obtain the optimal nonuniform subsampling probabilities, which can be calculated as

$$\eta_i^{\mathrm{opt}} = \frac{|\log\{f(\boldsymbol{x}_i|\hat{\theta})\}|}{\sum_{j=1}^{N}|\log\{f(\boldsymbol{x}_j|\hat{\theta})\}|}, \tag{2.4}$$

where $\hat{\theta}$ represents the maximum likelihood estimator (MLE), defined as

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log\{f(\boldsymbol{x}_i|\theta)\}.$$

The n random subsamples obtained using $\boldsymbol{\eta}^{\text{opt}} = (\eta_1^{\text{opt}}, \ldots, \eta_N^{\text{opt}})$ with replacement are called the most likely optimal (MLO) subsamples, and represented as $\{\boldsymbol{x}_1^{\text{MLO}}, \ldots, \boldsymbol{x}_n^{\text{MLO}}\}$. Then in each MH iteration, $l(\theta)$ can be approximated by

$$l_n^{\text{MLO}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log f(\boldsymbol{x}_i^{\text{MLO}}|\theta),$$

and $\Lambda(\theta, \theta')$ can be replaced with

$$\Lambda_n^{\text{MLO}}(\theta, \theta') = l_n^{\text{MLO}}(\theta') - l_n^{\text{MLO}}(\theta)$$

to speed up MCMC.

However, obtaining $\boldsymbol{\eta}^{\text{opt}}$ may not be easy since it requires evaluating the MLE $\hat{\theta}$ on the full data. In the next section, we introduce a novel method to reduce the full data into a set of optimal subsamples using energy distance, which is a statistical potential measure for the goodness-of-fit test between two samples.

3 Main results

3.1 The EDSS-MCMC algorithm

The key to the success of the subsampling MCMC is to accurately approximate $l(\theta)$ at different values of θ and θ' using the subsamples. In this paper, we aim to find the optimal subsamples $X_{U_n^*} = \{\boldsymbol{x}_i^{U_n^*}\}_{i=1}^n$ for a fixed $n \leqslant N$. Given any subsamples $X_{U_n} = \{\boldsymbol{x}_i^{U_n}\}_{i=1}^n$ from the full data $X = \{\boldsymbol{x}_i\}_{i=1}^N$, we can use

$$l_{U_n}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log f(\boldsymbol{x}_i^{U_n}|\theta)$$

to approximate $l(\theta)$. Now, we explain how to select the optimal subsamples based on the energy distance. Let F_N denote the empirical distribution function of the full data, and F_{U_n} represent the empirical distribution function of the subsamples X_{U_n}. We now introduce the definition of the energy distance between F_N and F_{U_n}.

Definition 3.1 (See [24]). The energy distance between F_N and F_{U_n} is defined as

$$\begin{aligned} ED(F_N, F_{U_n}) = {} & \frac{2}{nN} \sum_{i=1}^{n} \sum_{j=1}^{N} \|\boldsymbol{x}_i^{U_n} - \boldsymbol{x}_j\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{x}_i^{U_n} - \boldsymbol{x}_j^{U_n}\|_2 \\ & - \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \end{aligned}$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

This energy distance has the following property.

Proposition 3.2 (See [25]). *Suppose that $N \geqslant n > 1$. Then $ED(F_N, F_{U_n}) \geqslant 0$, and the equality holds if and only if $n = N$ and $X_{U_n} = X$.*

Proposition 3.2 suggests that $ED(F_N, F_{U_n})$ is always nonnegative, and it becomes small when F_{U_n} is close to F_N. Based on this proposition, the motivation for selecting the optimal subsamples through minimizing $ED(F_N, F_{U_n})$ relies on the following result.

Proposition 3.3. *Suppose that for all $\theta \in \Theta$, $\log f(\cdot|\theta)$ is an integrand function. Let $\tau(\theta) = \|\log f(\cdot|\theta)\|_{l_2}$. Then for any $X_{U_n} \subseteq X$, we have*

$$|l(\theta) - l_{U_n}(\theta)| = \left|\frac{1}{N}\sum_{i=1}^{N}\log f(\boldsymbol{x}_i|\theta) - \frac{1}{n}\sum_{j=1}^{n}\log f(\boldsymbol{x}_j^{U_n}|\theta)\right| \leqslant \tau(\theta)\sqrt{ED(F_N, F_{U_n})},$$

where $\|g\|_{l_2} = (\int_{\mathcal{X}} g^2(\boldsymbol{x})d\boldsymbol{x})^{1/2}$.

Proposition 3.3 indicates that by choosing X_{U_n} that minimizes $ED(F_N, F_{U_n})$, we can obtain a more accurate approximation $l_{U_n}(\theta)$. This forms the main idea of the EDSS-MCMC algorithm in this paper. Next, we explore how to obtain the optimal subsamples $X_{U_n^*}$ by minimizing $ED(F_N, F_{U_n})$.

In fact, since $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \overset{\text{i.i.d.}}{\sim} F$, Mak and Joseph [14] defined a set of representative points of F called support points through minimizing the energy distance between F and the empirical distribution of a set of points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. The optimization problem for obtaining support points is given by

$$\min_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n}\left(\frac{2}{nN}\sum_{i=1}^{n}\sum_{j=1}^{N}\|\boldsymbol{z}_i - \boldsymbol{x}_j\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2\right). \tag{3.1}$$

Mak and Joseph [14] developed an efficient algorithm based on difference-of-convex programming technique to generate the support points of F, which is implemented in the R package "support". However, these support points may not be a subset of the full data X. This is because the optimization in (3.1) is performed over a continuous space. In order to obtain the optimal subsamples $X_{U_n^*} = \{\boldsymbol{x}_i^{U_n^*}\}_{i=1}^n$, we need to solve the following discrete optimization problem:

$$\begin{aligned} X_{U_n^*} &= \arg\min_{X_{U_n}=\{\boldsymbol{x}_i^{U_n}\}_{i=1}^n\subseteq X} ED(F_N, F_{U_n}) \\ &= \arg\min_{X_{U_n}=\{\boldsymbol{x}_i^{U}\}_{i=1}^n\subseteq X}\left(\frac{2}{nN}\sum_{i=1}^{n}\sum_{j=1}^{N}\|\boldsymbol{x}_i^{U_n} - \boldsymbol{x}_j\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{x}_i^{U_n} - \boldsymbol{x}_j^{U_n}\|_2\right). \end{aligned} \tag{3.2}$$

However, using (3.2) directly to obtain the optimal subsamples may be time-consuming. In practice, we can first find the support points in a continuous space using (3.1) with the R package "support", which is a fast approach. Then we can select the optimal subsamples $X_{U_n^*}$ as the points in X that are closest to these support points based on the Euclidean distance. This can be done efficiently, even for large-scale datasets, using the K-dimensional (KD)-Tree based nearest neighbor algorithm.

However, it is important to acknowledge that the computational complexity of generating the support points of F using (3.1) is $O(N^2p)$. Therefore, if the size of the full data X is significantly large, obtaining the support points of F through (3.1) can be a time-consuming process. One possible solution is to split the space of samples $\mathcal{X}$ into $m = r^p$ grid-cells, where each cell has a side length of $1/r$, then obtain the optimal subsamples from each grid-cell, and finally merge these subsamples together. Specifically, let X_{E_k} denote the subsamples that fall into the k-th grid-cell, $k = 1, \ldots, m$. We can then obtain the optimal subsamples $X_{U_k^*} = \{\boldsymbol{x}_i^{U_k^*}\}_{i=1}^{n_k}$ of X_{E_k} by leveraging the R package "support" and the KD-Tree algorithm. Here, $n_k = \lceil |X_{E_i}|n/N\rceil$, where $\lceil x\rceil$ represents the minimum integer greater than x, and $|X_{E_i}|$ denotes the number of samples in X_{E_i}. Finally, we can obtain the approximate optimal subsamples by taking the union of $X_{U_k^*}$ for $k = 1, \ldots, m$, i.e., $X_{U_n^*} = \bigcup_{k=1}^m X_{U_k^*}$. Obviously, the selection of r will affect the final $X_{U_n^*}$. Based on the suggestion of r in [26] and our numerical simulation experience, we recommend using $r = \lfloor N^{1/(p+3)}\rfloor$. It should be noted that due to rounding $|X_{E_i}|n/N$ up to the nearest integer above, $|X_{E_i}|n/N$ may result in $\sum_{k=1}^m n_k$ exceeding the subsample size n. In practical applications, a common practice is to randomly discard surplus subsamples from $X_{U_n^*}$.

Figure 2 shows 100 subsamples based on the energy distance from a set of 1,000 full data points. In Figure 2(a), the grid-cell splitting technique is not utilized, while in Figure 2(b), the grid-cell splitting technique is employed. It is evident that both the energy distance-based subsamples provide better coverage of the full data compared with the subsamples generated with uniform probabilities and nonuniform probabilities, as depicted in Figure 1.

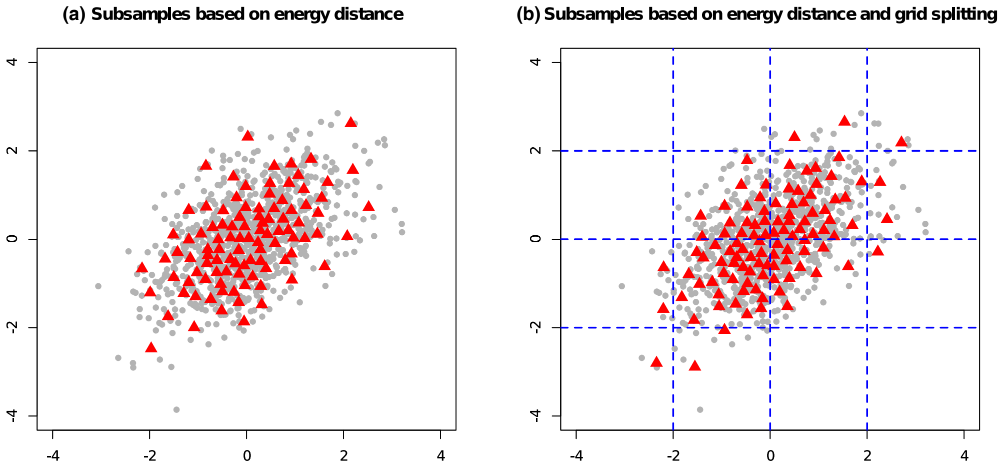


Figure 2 (Color online) 100 subsamples from 1,000 full data based on the energy distance, where the subsamples are denoted by triangles

After obtaining the optimal subsamples $X_{U_n^*}$ using the two aforementioned energy distance-based methods, we can utilize $Nl_{U_n^*}(\theta)$ to approximate the log-likelihood of the full data $Nl(\theta)$ in each iteration of the standard MH algorithm. Moreover, we can approximate the average log-likelihood ratio $\Lambda(\theta,\theta')$ using $\Lambda_{U_n^*}(\theta,\theta')=l_{U_n^*}(\theta')-l_{U_n^*}(\theta)$. Similar to (2.2), we can formally define the approximate acceptance probability as

$$\alpha_{U_n^*}(\theta,\theta')=1\wedge a_n^*(\theta,\theta'),\quad a_{U_n^*}(\theta,\theta')=\frac{p(\theta')\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta')^{N/n}q(\theta|\theta')}{p(\theta)\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta)^{N/n}q(\theta'|\theta)}. \tag{3.3}$$

Hence, in each iteration of the MH algorithm, when $u\sim U(0,1)$, $u<a_{U_n^*}(\theta,\theta')$ is equivalent to $\Lambda_{U_n^*}(\theta,\theta')>\psi(u,\theta,\theta')$. Consequently, the EDSS-MCMC can be described by Algorithms 2 and 3. Algorithm 2 applies when the full data is not split into m grid cells, whereas Algorithms 3 is designed for situations, where the full data is split into m grid cells to enhance the efficiency of Algorithm 2.

```
Algorithm 2: Energy distance-based subsampling MCMC
Input: $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N$, $q(\cdot\mid\theta)$
Output: $\theta_1,\ldots,\theta_T$
Obtain $X_{U_n^*}$ based on the R package “support” and the KD-Tree algorithm;
for $k=1,\ldots,T$ do
    $\theta\leftarrow\theta_k$;
    $\theta'\sim q(\cdot|\theta)$, $u\sim U(0,1)$;
    $\psi(u,\theta,\theta')\leftarrow\frac{1}{N}\log[u\frac{p(\theta)q(\theta'|\theta)}{p(\theta')q(\theta|\theta')}]$;
    $\Lambda_{U_n^*}(\theta,\theta')\leftarrow\frac{1}{n}\sum_{i=1}^n\log[\frac{f(\boldsymbol{x}_i^{U_n^*}|\theta')}{f(\boldsymbol{x}_i^{U_n^*}|\theta)}]$;
    if $\Lambda_{U_n^*}(\theta,\theta')>\psi(u,\theta,\theta')$ then
        $\theta_{k+1}=\theta'$ {Accept};
    else
        $\theta_{k+1}=\theta$ {Reject};
    end
end
```

3.2 Theoretical analysis of EDSS-MCMC

In this subsection, we present the upper bound on the total variation norm between the invariant distribution produced by the EDSS-MCMC algorithm and the target distribution π. This is a general theoretical framework that is applicable to both Algorithms 2 and 3. To begin, we introduce some essential concepts.

Algorithm 3: Grid- and energy distance-based subsampling MCMC

```
Input: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, $q(\cdot \mid \theta)$
Output: $\theta_1, \ldots, \theta_T$
Split the sample space $\mathfrak{X}$ into $m = r^d$ grid-cells with a side length of $1/r$, and let $X_{E_k}$ be the $k$-th subsample set which fall into the $k$-th grid-cell for $k = 1, \ldots, m$;
Let $X_{U_n^*} = \emptyset$;
for $k = 1, \ldots, m$ do
    Obtaining the optimal subsamples $X_{U_k^*} = \{\boldsymbol{x}_i^{U_k^*}\}_{i=1}^{n_k}$ of $X_{E_k}$ based on the R package "support" and the KD-Tree algorithm;
    $X_{U_n^*} = X_{U_n^*} \cup X_{U_k^*}$;
end
If $|X_{U_n^*}| > n$, we randomly delete $|X_{U_n^*}| - n$ extra subsamples from $X_{U_n^*}$; else, $X_{U_n^*}$ remains unchanged;
for $k = 1, \ldots, T$ do
    $\theta \leftarrow \theta_k$;
    $\theta' \sim q(\cdot|\theta)$, $u \sim U(0,1)$;
    $\psi(u, \theta, \theta') \leftarrow \frac{1}{N} \log[u \frac{p(\theta) q(\theta'|\theta)}{p(\theta') q(\theta|\theta')}]$;
    $\Lambda_{U_n^*}(\theta, \theta') \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log[\frac{f(\boldsymbol{x}_i^{U_n^*}|\theta')}{f(\boldsymbol{x}_i^{U_n^*}|\theta)}]$;
    if $\Lambda_{U_n^*}(\theta, \theta') > \psi(u, \theta, \theta')$ then
        $\theta_{k+1} = \theta'$ {Accept};
    else
        $\theta_{k+1} = \theta$ {Reject};
    end
end
```

Recall that the total variation norm between two distributions P and Q is defined as

$$\|P - Q\|_{TV} = 1/2 \int_{\Theta} |f_P(\theta) - f_Q(\theta)| d\theta,$$

where $f_P(\cdot)$ and $f_Q(\cdot)$ represent their respective densities. In the following part of this subsection, we establish that for a sufficiently large subsample size n, the EDSS-MCMC algorithm admits an invariant distribution.

Let K denote the Markov transition kernel of the standard MH chain targeted at π with a proposal q and the acceptance probability $\alpha(\theta, \theta')$ defined in (2.2). Then we have

$$K(\theta, d\theta') = \alpha(\theta, \theta') q(\theta'|\theta) d\theta' + \delta_\theta(d\theta') \rho(\theta),$$

where δ_θ is the Dirac delta function at θ, and $\rho(\theta) = 1 - \int_{\Theta} \alpha(\theta, \theta') q(\theta'|\theta) d\theta'$. After obtaining the optimal subsamples using the energy distance, whether through Algorithm 2 or Algorithm 3, we denote the final optimal subsamples by U_n^*. Let $K_{U_n^*}$ denote the Markov transition kernel of the Markov chain produced by EDSS-MCMC (Algorithm 2 or Algorithm 3) with a proposal q and approximate acceptance probability $\alpha_{U_n^*}(\theta, \theta')$ defined in (3.3). Then

$$K_{U_n^*}(\theta, d\theta') = \alpha_{U_n^*}(\theta, \theta') q(\theta'|\theta) d\theta' + \delta_\theta(d\theta') \rho_{U_n^*}(\theta),$$

where $\rho_{U_n^*}(\theta) = 1 - \int_{\Theta} \alpha_{U_n^*}(\theta, \theta') q(\theta'|\theta) d\theta'$. Furthermore, let K^i be the i-th step transition kernel targeted at π for $i \geqslant 2$ through $K^i(\theta, \cdot) = \int_{\Theta} K(\theta, \theta') K^{i-1}(\theta', \cdot) d\theta'$ with $K^1 = K$. We assume that K satisfies the following conditions:

(K.1) Geometric ergodicity. There exist a constant $\varrho \in (0,1)$ and a function $C : \Theta \to R^+$ such that for all $\theta \in \Theta$,

$$\|\pi - K^i(\theta, \cdot)\|_{TV} \leqslant C(\theta) \varrho^i.$$

(K.2) Uniform ergodicity. There exist two finite constants $C < \infty$ and $\varrho \in (0,1)$ such that

$$\sup_{\theta \in \Theta} \|\pi - K^i(\theta, \cdot)\|_{TV} \leqslant C \varrho^i.$$

Here, Condition (K.1) is milder than (K.2), and both Conditions (K.1) and (K.2) are grounded in commonly accepted assumptions in the literature [1, 2, 15]. These assumptions guarantee the consistency and convergence of the EDSS-MCMC algorithm.

Similar to K^i, let $K^i_{U^*_n}$ be the i-th iteration kernel corresponding to $K_{U^*_n}$, defined as

$$K^i_{U^*_n}(\theta,\cdot)=\int_\Theta K_{U^*_n}(\theta,\theta')K^{i-1}_{U^*_n}(\theta',\cdot)d\theta'$$

for $i\geqslant 2$. The following result demonstrates that for a sufficiently large subsample size n, $K_{U^*_n}$ is geometrically ergodic, and we denote its invariant distribution by π^*_n.

Theorem 3.4. *Under the assumptions of Condition* (K.1) *and* $\sup_{\theta\in\Theta}\rho(\theta)<1/2$*, there exists an* $n_0\leqslant N$ *such that for all* $n>n_0$*,* $K_{U^*_n}$ *is geometrically ergodic. In other words, there exist a constant* $\varrho\in(0,1)$ *and a function* $C:\Theta\to R^+$ *such that for all* $\theta\in\Theta$*,*

$$\|\pi^*_n-K^i_{U^*_n}(\theta,\cdot)\|_{TV}\leqslant C(\theta)\varrho^i.$$

In addition to admitting an invariant distribution for a sufficiently large n for the transition kernel $K_{U^*_n}$, we now demonstrate that under certain assumptions, it is possible to quantify the upper bound on the total variation norm between π and π^*_n. Our main result is inspired by the work of Alquier et al. [1] in the context of the EDSS-MCMC algorithm.

Theorem 3.5. *Define* $A_n=\sup_{\theta\in\Theta}1/\phi_{U^*_n}(\theta)$ *and*

$$B_n(\theta)=\int_\Theta q(\theta'|\theta)\alpha_{U^*_n}(\theta,\theta')|\phi_{U^*_n}(\theta)-\phi_{U^*_n}(\theta')|d\theta',$$

where

$$\phi_{U^*_n}(\theta)=\prod_{j=1}^n f(\boldsymbol{x}_j^{U^*_n}|\theta)^{N/n}\Big/\prod_{i=1}^N f(\boldsymbol{x}_i|\theta).$$

Assume that Conditions (K.1) *and* (K.2) *and* $\sup_{\theta\in\Theta}\rho(\theta)<1/2$ *hold. Then there exists a constant* $\kappa<\infty$ *such that*

$$\|K^i(\theta,\cdot)-K^i_{U^*_n}(\theta,\cdot)\|_{TV}\leqslant\kappa A_n\sup_{\theta\in\Theta}\ B_n(\theta)$$

and

$$\lim_{i\to\infty}\sup_{\theta\in\Theta}\|\pi-K^i_{U^*_n}(\theta,\cdot)\|_{TV}=\kappa A_n\sup_{\theta\in\Theta}\ B_n(\theta).$$

Furthermore, for a large enough subsample size n*, the invariant distribution* π^*_n *of the Markov chain produced by the EDSS-MCMC algorithm satisfies*

$$\|\pi-\pi^*_n\|_{TV}\leqslant\kappa A_n\sup_{\theta\in\Theta}\ B_n(\theta).$$

According to Proposition 3.3, we can easily obtain that $\phi_{U^*_n}(\theta)$ is controlled by $ED(F_N,F_{U^*_n})$. Hence, we can readily obtain the following corollary.

Corollary 3.6. *Define*

$$D_n(\theta)=\int_\Theta q(\theta'|\theta)\alpha_{U^*_n}(\theta,\theta')d\theta'\quad\textit{and}\quad\tau(\theta)=\|\log f(\cdot|\theta)\|_{l_2}.$$

Assume that Conditions (K.1) *and* (K.2) *and* $\sup_{\theta\in\Theta}\rho(\theta)<1/2$ *hold. Then there exists a constant* $\kappa<\infty$ *such that*

$$\|K^i(\theta,\cdot)-K^i_{U^*_n}(\theta,\cdot)\|_{TV}\leqslant 2\kappa\sup_{\theta\in\Theta}\exp\{2N\tau(\theta)\sqrt{ED(F_N,F_{U^*_n})}\}D_n(\theta)\tag{3.4}$$

and

$$\lim_{i\to\infty}\sup_{\theta\in\Theta}\|\pi-K^i_{U^*_n}(\theta,\cdot)\|_{TV}=2\kappa\sup_{\theta\in\Theta}\exp\{2N\tau(\theta)\sqrt{ED(F_N,F_{U^*_n})}\}D_n(\theta).\tag{3.5}$$

Furthermore, for a large enough subsample size n, the invariant distribution π_n^* of the Markov chain produced by the EDSS-MCMC algorithm satisfies

$$\|\pi-\pi_n^*\|_{TV}\leqslant 2\kappa\sup_{\theta\in\Theta}\exp\{2N\tau(\theta)\sqrt{ED(F_N,F_{U_n^*})}\}D_n(\theta). \tag{3.6}$$

Corollary 3.6 reveals that the total variation norm between π and π_n^* is determined by $E(F_N, F_{U_n^*})$. Thus, the smaller the value of $ED(F_N, F_{U_n^*})$, the smaller the total variation norm between π and π_n^*. This observation further indicates that the optimal subsamples should be chosen to minimize $ED(F_N, F_{U_n})$. All proofs are outlined in Appendix A.

Next, we provide a theoretical guarantee for the superiority of the proposed EDSS-MCMC algorithm compared with the Uniform-MCMC and MLO-MCMC algorithms. We know, to save computational efforts, the idea of subsampling MCMC is using only a Monte Carlo approximation of $\Lambda(\theta,\theta')$ based on a subset of the full data to decide whether (2.3) holds. Therefore, the performance of the subsampling MCMC algorithm can be measured by the absolute bias of the approximation of $\Lambda(\theta,\theta')$. The following conclusion provides an upper bound on the absolute bias of $\Lambda(\theta,\theta')$ under the three different subsampling methods in each iteration process.

Remark 3.7. For all $\theta,\theta'\in\Theta$, let $\Lambda_n^{\mathrm{Uni}}(\theta,\theta')$, $\Lambda_n^{\mathrm{MLO}}(\theta,\theta')$ and $\Lambda_n^{\mathrm{EDSS}}(\theta,\theta')$ denote the approximations of $\Lambda(\theta,\theta')$ under the three distinct subsampling techniques: Uniform-MCMC, MLO-MCMC and EDSS-MCMC algorithms, respectively. We can obtain

(i) $E|\Lambda_n^{\mathrm{Uni}}(\theta,\theta')-\Lambda(\theta,\theta')|\leqslant\frac{1}{N}\sum_{j=1}^N|\log\frac{f(\boldsymbol{x}_j|\theta')}{f(\boldsymbol{x}_j|\theta)}-\Lambda(\theta,\theta')|$;

(ii) $E|\Lambda_n^{\mathrm{MLO}}(\theta,\theta')-\Lambda(\theta,\theta')|\leqslant\sum_{j=1}^N|\frac{1}{N}\log\frac{f(\boldsymbol{x}_j|\theta')}{f(\boldsymbol{x}_j|\theta)}-\eta_j^{\mathrm{opt}}\Lambda(\theta,\theta')|$, where η_j^{opt} is defined in (2.4);

(iii) $|\Lambda_n^{\mathrm{EDSS}}(\theta,\theta')-\Lambda(\theta,\theta')|\leqslant 2\tau_{\max}\sqrt{ED(F_N,F_{U_n^*})}$, where

$$\Lambda_n^{\mathrm{EDSS}}(\theta,\theta')=\frac{1}{n}\sum_{i=1}^n\log\left[\frac{f(\boldsymbol{x}_i^{U_n^*}\mid\theta')}{f(\boldsymbol{x}_i^{U_n^*}\mid\theta)}\right],\quad \tau_{\max}=\sup_{\theta\in\Theta}\tau(\theta),$$

and $X_{U_n^*}=\{\boldsymbol{x}_i^{U_n^*}\}_{i=1}^n$ is the optimal subsamples obtained by the EDSS-MCMC algorithm.

Since $\tau_{\max}$ is always bounded and the definition of U_n^* in (3.2) means that $ED(F_N, F_{U_n^*})$ is very close to zero, $|\Lambda_n^{\mathrm{EDSS}}(\theta,\theta')-\Lambda(\theta,\theta')|$ is always smaller than both $E|\Lambda_n^{\mathrm{Uni}}(\theta,\theta')-\Lambda(\theta,\theta')|$ and $E|\Lambda_n^{\mathrm{MLO}}(\theta,\theta') - \Lambda(\theta,\theta')|$. This remark theoretically demonstrates the superiority of the proposed EDSS-MCMC algorithm over the other two subsampling methods.

4 Simulation

In this section, we conduct numerical experiments to assess the performance of the proposed EDSS-MCMC algorithm presented in Algorithm 3. We compare its performance with that of the standard MH algorithm, referred to as Full-MCMC, as well as two subsampling MH algorithms: Uniform-MCMC and MLO-MCMC. The simulations are conducted on a computer equipped with an Intel Core i7-8700T CPU operating at 2.40 GHz. Additionally, the system has 8 GB of RAM.

4.1 Inference of an AR(1) model

We begin by testing the proposed method on an autoregressive time series $\{Y_k, k\leqslant N\}$ following an AR(1) model, defined recursively as

$$\begin{cases}Y_1\sim N(0,1),\\ Y_k=\theta_1+\theta_2Y_{k-1}+\epsilon_k,\quad \epsilon_k\sim N(0,1),\quad k=2,\ldots,N.\end{cases} \tag{4.1}$$

This model has also been used in [22] to demonstrate the efficiency of the subsampling MH algorithm. Following their setup, we use the same true parameter values $\theta^*=(0.3,0.6)$ and the same prior distribution

$$p(\theta_1,\theta_2)\overset{\text{ind.}}{=}\mathcal{U}(\theta_1|-5,5)\mathcal{U}(\theta_2|0,1),$$

where $\mathcal{U}(\cdot|a,b)$ denotes the uniform density on the interval $[a,b]$. Additionally, we utilize the proposal distribution $q(\theta|\theta_c)=\mathcal{N}(\theta|\theta_c,\Sigma_c)$, where Σ_c is the negative inverse Hessian matrix of the log-posterior evaluated at θ_c. We generate a time series $\{Y_k, k\leqslant N\}$ of length $N=10^6$ according to the AR(1) model in (4.1) with the true parameter values θ^*.

We begin by highlighting the advantages of using energy distance for selecting representative subsamples. As demonstrated in Remark 3.7, the core principle of subsampling MCMC involves employing subsamples to estimate the average log-likelihood ratio, $\Lambda(\theta,\theta')$. We evaluate the subsamples' representativeness by examining the absolute bias of the approximate estimator. We repeat the three subsampling methods 100 times under three different subsample sizes n to obtain approximate estimators for $\Lambda(\theta,\theta')$. Figure 3 depicts the boxplots of absolute biases for the evaluated estimators with "EDSS", "MLO" and "Uni" denoting the EDSS-MCMC, MLO-MCMC and Uniform-MCMC algorithms, respectively. Theoretically, the EDSS-MCMC algorithm leads to $|\Lambda_n^{\mathrm{EDSS}}(\theta,\theta')-\Lambda(\theta,\theta')|$ decreasing to zero. However, the algorithm's implementation introduces randomness, causing a non-zero bias. Nevertheless, this bias decreases to zero as the number of subsamples increases, as depicted in Figure 3. Moreover, the figure indicates that the absolute biases in estimating $\Lambda(\theta,\theta')$ using MLO-MCMC and Uniform-MCMC algorithms are significantly greater than that using the EDSS-MCMC algorithm.

Next, in our simulations, we execute each MCMC algorithm for 5,000 iterations, discarding the first 1,000 iterations as burn-in. Then we use the remaining samples to estimate θ_1 and θ_2, and compare the CPU runtime, bias and empirical mean squared error.

We perform the simulation B times and calculate the empirical bias and mean squared error as follows:

$$\mathrm{Bias}=\left|\frac{1}{B}\sum_{b=1}^{B}\theta_b-\theta\right|,\quad \mathrm{MSE}=\frac{1}{B}\sum_{b=1}^{B}(\theta_b-\theta)^2. \tag{4.2}$$

Here, θ_b represents the estimator in the b-th repetition, and θ denotes the true value of the parameter.

Table 1 presents the diagnostic metrics averaged over $B=100$ repetitions of each algorithm with $n\in\{0.001N,0.005N,0.01N\}$. The results highlight several advantages of the EDSS-MCMC algorithm: (i) Compared with the Full-MCMC algorithm, the EDSS-MCMC algorithm significantly reduces the CPU runtime; (ii) In comparison to other subsampling MCMC algorithms, although the EDSS-MCMC algorithm has a slightly longer CPU runtime, it exhibits a smaller bias and mean squared error; (iii) As the subsample size increases, the bias and mean squared error of the EDSS-MCMC algorithm tends to become smaller.

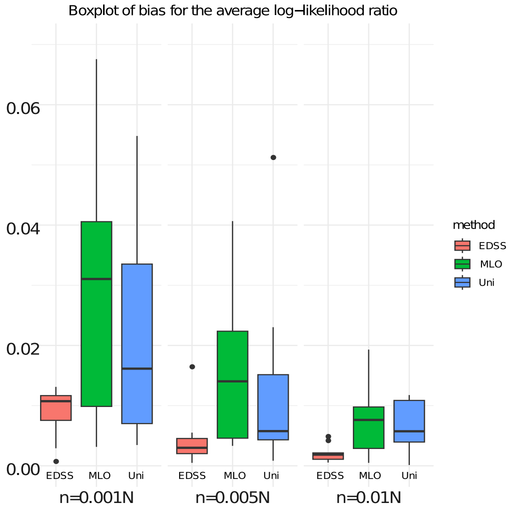
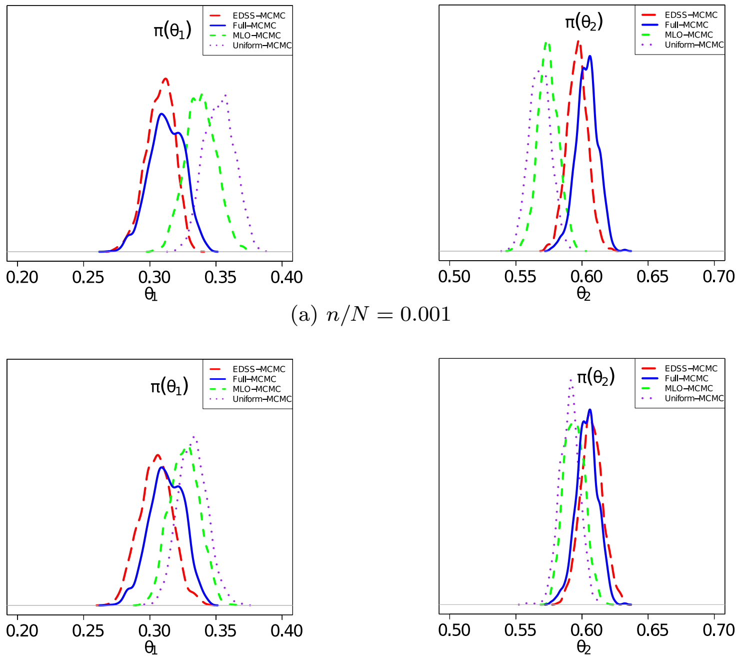


Figure 3 (Color online) Boxplots depicting the absolute biases of $\Lambda(\theta,\theta')$ for the AR(1) model under three sampling methods and three subsample sizes

Table 1 Diagnostic metrics for four MCMC algorithms under a fixed number of iterations for the AR(1) model

			θ_1		θ_2	
Subsample size (n)	Method	Time (s)	Bias	$\sqrt{\text{MSE}}$	Bias	$\sqrt{\text{MSE}}$
N	Full-MCMC	1648.12	0.0059	0.0061	0.0052	0.0048
$0.001N$	Uniform-MCMC	2.36	0.0785	0.1391	0.0703	0.1012
	MLO-MCMC	4.47	0.1060	0.1098	0.0636	0.0762
	EDSS-MCMC	6.78	0.0593	0.0699	0.0386	0.0445
$0.005N$	Uniform-MCMC	10.23	0.0591	0.0848	0.0411	0.0613
	MLO-MCMC	18.52	0.0321	0.0373	0.0174	0.0206
	EDSS-MCMC	22.91	0.0135	0.0165	0.0097	0.0130
$0.01N$	Uniform-MCMC	27.91	0.0291	0.0127	0.0221	0.0193
	MLO-MCMC	49.86	0.0102	0.0014	0.0126	0.0058
	EDSS-MCMC	95.55	0.0087	0.0003	0.0007	0.0019



(a) $n/N = 0.001$

(b) $n/N = 0.005$

Figure 4 (Color online) Kernel density estimates of marginal posteriors obtained by four MCMC algorithms, each with 5,000 iterations and the first 1,000 iterations being discarded as burn-in for $n \in \{0.001N, 0.005N\}$ under the AR(1) model

Next, we examine the influence of three subsampling MCMC algorithms on the marginal distribution compared with the Full-MCMC algorithm across various subsample sizes, as shown in Figure 4. In our simulations, we execute each algorithm for 5,000 iterations, discarding the first 1,000 iterations as burn-in, and then use the remaining samples to estimate the kernel density of marginal posteriors. Figure 4 shows that among these three subsampling MCMC algorithms, the EDSS-MCMC algorithm provides a better approximation closer to the Full-MCMC algorithm.

4.2 Binary classification

We consider a training dataset consisting of $N = 10^7$ labeled observations $Y = \{Y_k, k = 1, \ldots, N\}$ from a Gaussian mixture model, which is simulated with

$$Y_k | I_k = i \sim N(\theta_i, 1), \quad I_k \sim \text{Bernoulli}(1/2),$$

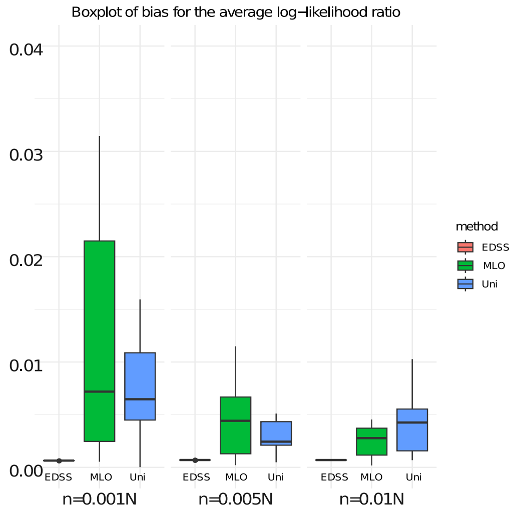


Figure 5 (Color online) Boxplots depicting the absolute biases of $\Lambda(\theta,\theta')$ for the binary classification model under three sampling methods and three subsample sizes

where $i \in \{1,2\}$, and the true values are $\theta_1^* = -1$ and $\theta_2^* = 1$. Similar to Maire et al. [15], we choose the prior $p(\theta_1,\theta_2) \overset{\text{ind.}}{=} N(0,1/2)\Gamma(1,2)$, where $\Gamma(a,b)$ is the Gamma distribution with shape a and rate b.

In the context of binary classification, we replicate the analysis from Figure 3, employing three subsampling methods to estimate $\Lambda(\theta,\theta')$ across three different subsample sizes. To ensure statistical robustness, we perform this procedure 100 times and present the resulting boxplots of the absolute biases for these estimators in Figure 5, where the symbols “EDSS”, “MLO” and “Uni” retain the same meanings as those employed in Figure 3. This figure further reflects the significant advantage of selecting representative subsamples based on the energy distance.

Next, we consider using the classification error rate [15] on a test dataset $Y^* = \{Y_k^*, k = 1,\dots,N_{\text{test}}\}$ ($N_{\text{test}} = 10^4$) to evaluate the performance of the algorithm, which is defined as $\epsilon = \|I^* - I_{\text{true}}\|_2$, where $I_{\text{true}} = \{I_{\text{true},1},\dots,I_{\text{true},\text{N}_{\text{test}}}\}$ is the true class of Y^*. Based on the four algorithms, i.e., the Full-MCMC, Uniform-MCMC, MLO-MCMC and EDSS-MCMC algorithms, we can obtain the estimator $\hat{\theta} = (\hat{\theta}_1,\hat{\theta}_2)$, and then obtain the predicted classification labels $I^* = \{I_1^*,\dots,I_{N_{\text{test}}}^*\}$, where

$$I_k^* = \arg\max_{i\in\{1,2\}} f(Y_k^*|\hat{\theta}_i), \quad k = 1,\dots,N_{\text{test}},$$

and $f(\cdot|\hat{\theta}_1)$ and $f(\cdot|\hat{\theta}_2)$ denote the densities for Gaussian distributions $N(\hat{\theta}_1,1)$ and $N(\hat{\theta}_2,1)$, respectively.

Table 2 Classification errors for four MCMC algorithms with 5,000 iterations

Subsample size (n)	Method	Time (s)	Mean	25%	50%	50%
N	Full-MCMC	1925.09	0.1639	0.1628	0.1641	0.1650
$0.001N$	Uniform-MCMC	0.62	0.5426	0.1809	0.8185	0.8198
	MLO-MCMC	2.75	0.5668	0.2648	0.8345	0.8648
	EDSS-MCMC	4.78	0.1721	0.1648	0.1650	0.1652
$0.005N$	Uniform-MCMC	1.36	0.5430	0.1821	0.8178	0.8185
	MLO-MCMC	10.72	0.5717	0.1721	0.8250	0.8352
	EDSS-MCMC	19.70	0.1688	0.1648	0.1650	0.1651
$0.01N$	Uniform-MCMC	9.31	0.4579	0.1817	0.1848	0.8146
	MLO-MCMC	21.58	0.4046	0.1645	0.1750	0.1845
	EDSS-MCMC	39.69	0.1643	0.1640	0.1644	0.1649

We conduct 5,000 iterations for each of the four algorithms, discarding the initial 1,000 samples for burn-in. The remaining samples are employed for parameter estimation, followed by classification error computation. This process is replicated 100 times, and the results, presented in Table 2, include the average runtimes, and the mean, 25%, 50% and 75% quantiles of the classification errors. The findings indicate that the EDSS-MCMC algorithm yields lower classification errors compared with Uniform-MCMC and MLO-MCMC. Furthermore, when n reaches $0.01N$, EDSS-MCMC demonstrates comparable performance to Full-MCMC with substantial computational time savings.

4.3 Logistic regression

Consider a d-dimensional logistic regression model parameterized by a vector $\theta = (\theta_1, \ldots, \theta_d) \in \Theta$. The observations are generated according to the following two steps:

(i) simulating the covariates: generating the covariate vector $X_i = (X_{i,1}, \ldots, X_{i,p})$ from a multivariate normal distribution $X_i \sim \mathcal{N}(0, (1/p)^2 I_p)$, where I_p is a $p \times p$ identity matrix;

(ii) given θ and X_i, generating the response Y_i independently from a Bernoulli distribution with the parameter $\gamma_i = 1/(1 + \mathrm{e}^{-\theta X_i^T})$.

In the simulation, we generate a dataset consisting of $N = 10^6$ observations under the true parameter values $\theta^* = (1, 2, -1)$ for $d = 3$ in the logistic regression model. The prior distribution for θ is specified as a zero-mean Gaussian distribution with a covariance matrix $\Sigma_\theta = 10 I_d$.

In the logistic regression model, we first conduct analyses similar to that of Figures 3 and 5. Figure 6 presents the boxplots of the absolute biases of $\Lambda(\theta, \theta')$ following 100 repetitions under three different subsample sizes, employing three subsampling methods, where the symbols "EDSS", "MLO" and "Uni" retain the same meanings as those employed in Figure 3. We can draw a similar conclusion as that of Figures 3 and 5. This demonstrates that the EDSS-MCMC algorithm outperforms the other two methods significantly.

Next, we compare the various MCMC algorithms with subsample sizes $n \in \{0.001N, 0.005N, 0.01N\}$. For each algorithm, we run 5,000 iterations and discard the first 1,000 iterations as burn-in. Table 3 presents the diagnostic metrics obtained by running each algorithm 100 times. These metrics include the CPU runtime, bias and empirical mean squared error for θ_1, θ_2 and θ_3. Here, the definitions of bias and empirical mean squared error are the same as in (4.2). As expected, the Full-MCMC produces more accurate estimators of θ_1, θ_2 and θ_3. However, compared with the Full-MCMC algorithm, all the subsampling MCMC algorithms are significantly faster. Furthermore, among the subsampling MCMC

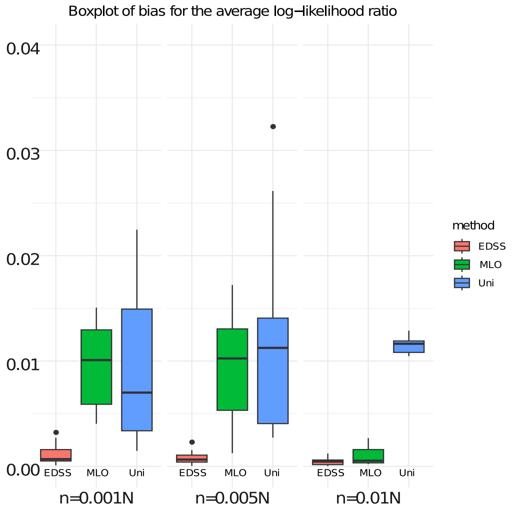
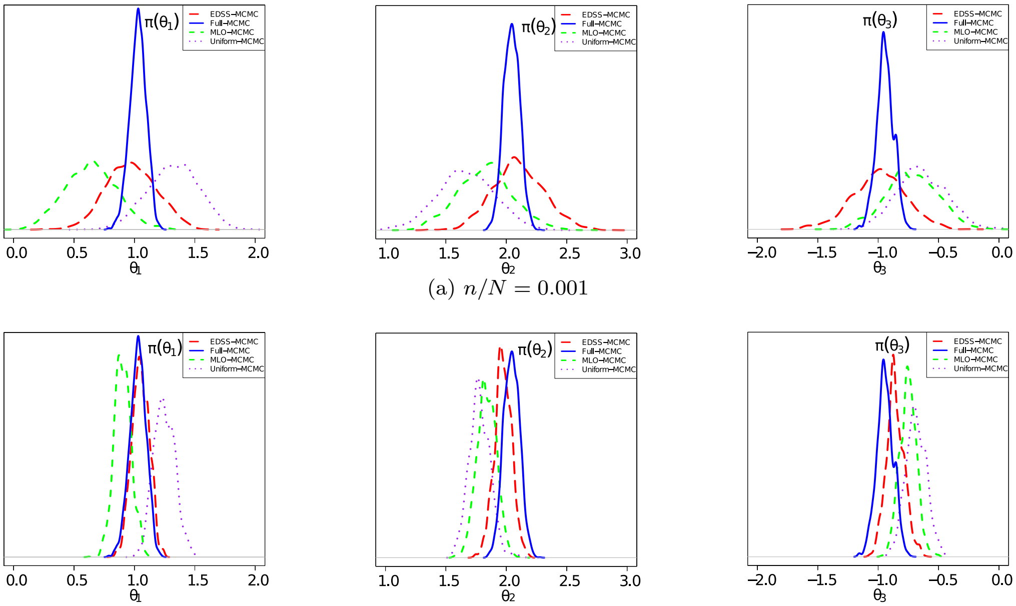


Figure 6 (Color online) Boxplots depicting the absolute bias of average likelihood ratio estimation for the logistic regression model under three sampling methods and three sample sizes

Table 3 Diagnostic metrics for four MCMC algorithms under a fixed number of iterations for the logistic regression model

			θ_1		θ_2		θ_3	
Subsample size (n)	Method	Time (s)	Bias	$\sqrt{\text{MSE}}$	Bias	$\sqrt{\text{MSE}}$	Bias	$\sqrt{\text{MSE}}$
N	Full-MCMC	6073.08	0.0317	0.0013	0.0194	0.0038	0.0334	0.0044
$0.001N$	Uniform-MCMC	6.67	0.2294	0.0731	0.2766	0.0905	0.2679	0.1076
	MLO-MCMC	10.52	0.1061	0.0192	0.1478	0.0306	0.0613	0.0164
	EDSS-MCMC	11.24	0.0858	0.0135	0.1011	0.0267	0.0498	0.0147
$0.005N$	Uniform-MCMC	11.48	0.0854	0.0439	0.0517	0.1633	0.0863	0.0101
	MLO-MCMC	21.63	0.0773	0.0143	0.0419	0.0102	0.0576	0.0082
	EDSS-MCMC	35.42	0.0508	0.0064	0.0384	0.0052	0.0413	0.0043
$0.01N$	Uniform-MCMC	25.67	0.0113	0.0162	0.0104	0.0142	0.0072	0.0078
	MLO-MCMC	55.31	0.0660	0.0067	0.0398	0.0065	0.0737	0.0160
	EDSS-MCMC	89.26	0.0103	0.0043	0.0130	0.0028	0.0106	0.0016



(a) $n/N = 0.001$

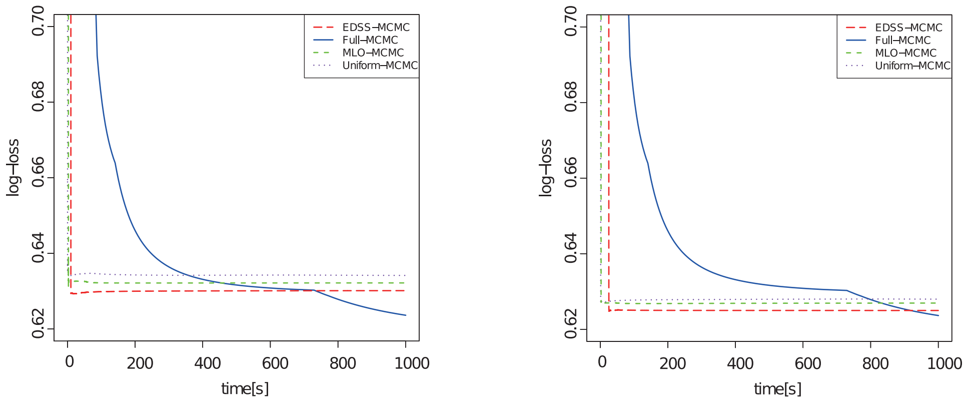
(b) $n/N = 0.005$

Figure 7 (Color online) Kernel density estimates of marginal posteriors obtained by four MCMC algorithms, each with 5,000 iterations and the first 1,000 iterations being discarded as burn-in for $n \in \{0.001N, 0.005N\}$ under the logistic regression model

algorithms, the EDSS-MCMC algorithm, while having a slightly longer CPU runtime, demonstrates the lower bias and mean squared error. This indicates that the EDSS-MCMC algorithm offers more accurate estimates in comparison to other subsampling MCMC algorithms.

As shown in Figure 4, Figure 7 displays the marginal posteriors of θ_1, θ_2 and θ_3 computed by four subsampling MCMC methods (Uniform-MCMC, MLO-MCMC, EDSS-MCMC and Full-MCMC) with different subsample sizes $n \in \{0.001N, 0.005N\}$. The Full-MCMC algorithm serves as a benchmark due to its use of the full data. Each curve represents the kernel density estimates of marginal posteriors from 5,000 iterations, discarding the initial 1,000 as burn-in. It is evident that the EDSS-MCMC method appears to outperform Uniform-MCMC and MLO-MCMC. Furthermore, as the subsample size n grows to $0.005N$, the marginal posterior kernel density estimates from the EDSS-MCMC algorithm are very close to those of the benchmark, i.e., the Full-MCMC algorithm. This suggests that the EDSS-MCMC algorithm is a competitive alternative to the Full-MCMC method for this specific logistic regression model.

In addition to the comparisons presented in Table 3 and Figure 7, the log-loss [20] is another popular



(a) $n/N = 0.001$

(b) $n/N = 0.005$

Figure 8 (Color online) Live log-losses of four MCMC algorithms on the same test set for the logistic regression model

metric for evaluating the predictive accuracy of a classifier on a held-out test dataset $T_* = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$. For the case of a binary response, the log-loss (lower is better) is defined as

$$l(\theta, T_*) = -\frac{1}{|T_*|} \sum_{(y_i, \boldsymbol{x}_i) \in T_*} \{y_i \log f(\boldsymbol{x}_i|\theta) + (1 - y_i) \log(1 - f(\boldsymbol{x}_i|\theta))\}.$$

For any estimator of θ, the log-loss can be utilized to assess the accuracy of the estimator.

For two different subsample sizes, we execute four MCMC algorithms, each run for $t_0 = 1{,}000$ seconds. The resulting chains are denoted by $\{\theta_t^{\text{Full-MCMC}}, 0 \leqslant t \leqslant t_0\}$, $\{\theta_t^{\text{Uniform-MCMC}}, 0 \leqslant t \leqslant t_0\}$, $\{\theta_t^{\text{MLO-MCMC}}, 0 \leqslant t \leqslant t_0\}$ and $\{\theta_t^{\text{EDSS-MCMC}}, 0 \leqslant t \leqslant t_0\}$, respectively. On a test set of size $m = 10^5$, denoted by T_*, we calculate the live log-losses corresponding to the chains $\{l(\theta_t^{\text{Full-MCMC}}, T_*), 0 \leqslant t \leqslant t_0\}$, $\{l(\theta_t^{\text{Uniform-MCMC}}, T_*), 0 \leqslant t \leqslant t_0\}$, $\{l(\theta_t^{\text{MLO-MCMC}}, T_*), 0 \leqslant t \leqslant t_0\}$ and $\{l(\theta_t^{\text{EDSS-MCMC}}, T_*), 0 \leqslant t \leqslant t_0\}$, respectively. A comparison of these live log-losses is presented in Figure 8.

As expected, the Full-MCMC algorithm is penalized because it evaluates the full data of size $N = 10^6$ at each iteration. For this model, the Uniform-MCMC and MLO-MCMC algorithms converge faster than the Full-MCMC algorithm, but they suffer from larger log-losses when the Markov chain reaches a stationary state. However, the EDSS-MCMC algorithm strikes a good balance between computational cost and accuracy. In fact, for $n = 0.005N$, the log-losses of the EDSS-MCMC algorithm are similar to those of the Full-MCMC algorithm, while converging at least 30 times faster. Consequently, the EDSS-MCMC algorithm proves to be a promising choice to accelerate the MCMC algorithm in this context.

5 Real data analysis

We consider two datasets to fit the logistic regression models. The first dataset is covtype.binary [4, 6] which was originally a classification problem with 7 classes. We follow Collobert et al. [6] and Bardenet et al. [4] and transform it into a binary classification problem. This dataset consists of 550,087 observations and $d = 11$ covariates (with an intercept). For our analysis, we choose a training subset consisting of $N = 200{,}000$ observations, and a separate testing subset T_* consisting of $m = 10{,}000$ observations. The prior of the parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_{10})$ is taken to be a Cauchy distribution which is recommended by Gelman et al. [8]. The second dataset concerns firm bankruptcy with 4,748,089 observations, where firm default is the response variable and there are eight firm-specific and macroeconomic covariates ($d = 9$, including an intercept) (see [9] for details). We select $N = 3{,}000{,}000$ observations for the train set and use the remaining observations as the test set, denoted by T_*. Similar to [22], we choose a Gaussian prior for the parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_8)$. We benchmark the proposed EDSS-MCMC algorithm (using

Table 4 Diagnostic metrics for four MCMC algorithms under a fixed computational budget $t_0 = 1{,}000\,\mathrm{s}$ and three different subsample sizes n for the two datasets

		Covtype.binary dataset		Bankruptcy dataset	
Subsample size (n)	Method	AMTV (↓)	log-loss (↓)	AMTV (↓)	log-loss (↓)
N	Full-MCMC	0.231	0.680	0.258	0.860
$0.001N$	Uniform-MCMC	0.418	0.735	0.320	0.893
	MLO-MCMC	0.397	0.711	0.251	0.929
	EDSS-MCMC	0.339	0.695	0.238	0.821
$0.005N$	Uniform-MCMC	0.371	0.705	0.201	0.517
	MLO-MCMC	0.341	0.694	0.235	0.641
	EDSS-MCMC	0.277	0.685	0.207	0.389
$0.01N$	Uniform-MCMC	0.252	0.698	0.198	0.492
	MLO-MCMC	0.243	0.687	0.195	0.433
	EDSS-MCMC	0.235	0.682	0.186	0.359

Algorithm 3) against the Full-MCMC, MLO-MCMC and Uniform-MCMC algorithms, and evaluate these four algorithms on the aforementioned datasets, imposing a consistent computational budget of $t_0 = 1{,}000$ seconds.

We consider two quantification metrics: the average of the marginal total variation (AMTV) norm and the log-loss. The AMTV (lower is better) is used to measure the estimation of the marginal distribution for $\theta = (\theta_1, \ldots, \theta_d)$, which is defined as

$$\mathrm{AMTV} = \frac{1}{d}\sum_{j=1}^{d} \|\pi^{(j)} - \tilde{\pi}^{(j)}\|_{TV},$$

where $\pi^{(j)}$ and $\tilde{\pi}^{(j)}$ are respectively the true j-th marginal and the j-th marginal of the chain distribution after a runtime of $t_0 = 1{,}000$ seconds, respectively. The true marginals are estimated from a long MH chain. Additionally, the log-loss is calculated for each algorithm using the same test set T_*. The log-losses are computed by averaging the final 1,000 posterior samples under each algorithm.

Table 4 provides the diagnostic metrics for each algorithm with three different subsample sizes n. For the covtype.binary dataset, the Full-MCMC algorithm achieves stationarity after the fixed runtime t_0. As the subsample size n increases, the performance of the three subsampling MCMC algorithms progressively converges to that of the Full-MCMC algorithm. However, compared with the other two subsampling MCMC algorithms, the EDSS-MCMC algorithm exhibits smaller AMTV values and log-losses. For the bankruptcy dataset, the Full-MCMC algorithm fails to attain stationarity within the fixed runtime t_0. In contrast, the three subsampling algorithms successfully reach a stationary state. Notably, the EDSS-MCMC algorithm exhibits smaller AMTV values and log-losses compared with the other two subsampling algorithms.

Next, we use the live log-loss to study the convergence of each algorithm. Similar to the live log-losses shown in Figure 8, Figure 9 also displays the live log-losses of the four MCMC algorithms on the same test dataset T_* with the top row corresponding to the covtype.binary dataset and the bottom row to the bankruptcy dataset. The left column corresponds to $n/N = 0.001$, and the right column corresponds to $n/N = 0.005$. The comparison reveals that the EDSS-MCMC algorithm exhibits faster convergence compared with the Full-MCMC method. Furthermore, when the chains reach stationarity, the EDSS-MCMC consistently yields lower log-losses compared with the other two subsampling MCMC algorithms.

6 Discussion

The MH algorithm is widely used for Bayesian inference, but it does not scale well to large-scale datasets when the computational budget is limited. In this paper, we propose an efficient subsampling MH

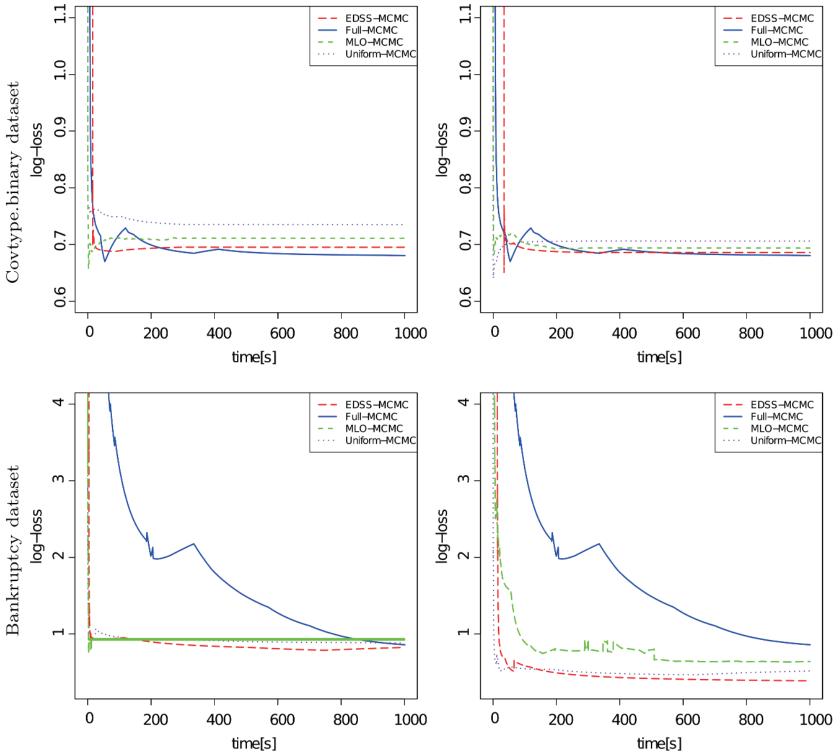


Figure 9 (Color online) Live log-losses of four MCMC algorithms on the test set T_* for the two datasets

algorithm called the EDSS-MCMC algorithm. This algorithm utilizes the energy distance to select representative subsamples of the full data, enabling faster computation. The energy distance, a statistical potential measure initially introduced by Székely and Rizzo [24] for testing goodness-of-fit. The significant advantage of using the energy distance to select representative subsamples is that it ensures that the resulting subsamples closely match the distribution of the full data over the Uniform-MCMC and MLO-MCMC algorithms, and then these representative subsamples enjoy an improved error rate in Monte Carlo estimates of the average log-likelihood ratio, $\Lambda(\theta,\theta')$ (as demonstrated in Remark 3.7). Additionally, we quantify the total variation norm between the invariant distribution produced by the EDSS-MCMC algorithm and the target distribution. The numerical experiments demonstrate that the proposed algorithm performs excellently compared with the standard MH algorithm and the other subsampling MCMC algorithms under a fixed computational budget.

While this paper introduces interesting subsampling MH algorithms based on the energy distance, there are several promising avenues for future research. Firstly, this paper assumes that the full data is available at once. However, developing a subsampling MCMC algorithm in an online learning setting would be a valuable and practical extension. Secondly, we must admit that the computational burden for obtaining representative subsamples by minimizing the energy distance will increase when the size or dimension of the full data increases. In the future, we hope to explore more effective techniques to alleviate the computational burden.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 12401324, 12131001, 12371259 and 12371260) and National Key Research and Development Program of China (Grant No. 2020YFA0714102). The authors thank the referees for their helpful comments and suggestions.

References

1 Alquier P, Friel N, Everitt R, et al. Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. Stat Comput, 2016, 26: 29–47

2 Andrieu C, Roberts G O. The pseudo-marginal approach for efficient Monte Carlo computations. Ann Statist, 2009, 37: 697–725

3 Andrieu C, Vihola M. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. Ann Appl Probab, 2015, 25: 1030–1077

4 Bardenet R, Doucet A, Holmes C. Towards scaling up Markov chain Monte Carlo: An adaptive subsampling approach. In: Proceedings of International Conference on Machine Learning. San Diego: JMLR, 2014, 405–413

5 Bardenet R, Doucet A, Holmes C. On Markov chain Monte Carlo methods for tall data. J Mach Learn Res, 2017, 18: 1515–1557

6 Collobert R, Bengio S, Bengio Y. A parallel mixture of SVMs for very large scale problems. Neural Comput, 2002, 14: 1105–1114

7 Gelfand A E, Smith A F M. Sampling-based approaches to calculating marginal densities. J Amer Statist Assoc, 1990, 85: 398–409

8 Gelman A, Jakulin A, Pittau M G, et al. A weakly informative default prior distribution for logistic and other regression models. Ann Appl Probab, 2008, 2: 1360–1383

9 Giordani P, Jacobson T, Schedvin E, et al. Taking the twists into account: Predicting firm bankruptcy risk with splines of financial ratios. J Financ Quant Anal, 2014, 49: 1071–1099

10 Hastings W K. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 1970, 57: 97–109

11 Hu G, Wang H. Most likely optimal subsampled Markov chain Monte Carlo. J Syst Sci Complex, 2021, 34: 1121–1134

12 Jacob P E, Thiery A H. On nonnegative unbiased estimators. Ann Statist, 2015, 43: 769–784

13 Korattikara A, Chen Y, Welling M. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In: Proceedings of International Conference on Machine Learning. San Diego: JMLR, 2014, 181–189

14 Mak S, Joseph V R. Support points. Ann Statist, 2018, 46: 2562–2592

15 Maire F, Friel N, Alquier P. Informed sub-sampling MCMC: Approximate Bayesian inference for large datasets. Stat Comput, 2019, 29: 449–482

16 Medina-Aguayo F J, Lee A, Roberts G O. Stability of noisy Metropolis-Hastings. Stat Comput, 2016, 26: 1187–1211

17 Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines. J Chem Phys, 1953, 21: 1087–1092

18 Meyn S P, Tweedie R L. Markov Chains and Stochastic Stability. Cambridge: Cambridge Univ Press, 2009

19 Mitrophanov A Y. Sensitivity and convergence of uniformly ergodic Markov chains. J Appl Probab, 2005, 142: 1003–1014

20 Nemeth C, Fearnhead P. Stochastic gradient Markov chain Monte Carlo. J Amer Statist Assoc, 2021, 116: 433–450

21 Nemeth C, Sherlock C. Merging MCMC subposteriors through Gaussian-process approximations. Bayesian Anal, 2018, 13: 507–530

22 Quiroz M, Kohn R, Villani M, et al. Speeding up MCMC by efficient data subsampling. J Amer Statist Assoc, 2019, 114: 831–843

23 Scott S L, Blocker A W, Bonassi F V, et al. Bayes and big data: The consensus Monte Carlo algorithm. Int J Manag Sci Eng Manag, 2016, 11: 78–88

24 Székely G J, Rizzo M L. Testing for equal distributions in high dimension. InterStat, 2004, 5: 1249–1272

25 Szekely G J, Rizzo M L. Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. J Classification, 2005, 22: 151–183

26 Zhao Y B, Amemiya Y S, Hung Y. Efficient Gaussian process modeling using experimental design-based subagging. Statist Sinica, 2018, 28: 1459–1479

Appendix A

This appendix provides proofs of Proposition 3.3, Theorems 3.4 and 3.5, Corollary 3.6 and Remark 3.7. We first present an important lemma.

Lemma A.1 (See [14]). *Let g be an integrand function, and $\{\boldsymbol{z}_i\}_{i=1}^n$ be the samples of distribution F with F_n as its empirical distribution function. Then we have*

$$\left| \int_{\mathcal{X}} g(\boldsymbol{x}) dF - \frac{1}{n} \sum_{j=1}^{n} g(\boldsymbol{z}_i) \right| \leqslant \|g\|_{l_2} \sqrt{E(F, F_n)},$$

where

$$E(F, F_n) = \frac{2}{n} \sum_{i=1}^{n} \mathbb{E}\|\boldsymbol{z}_i - \boldsymbol{Y}\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 - \mathbb{E}\|\boldsymbol{Y} - \boldsymbol{Y}'\|_2$$

and $\boldsymbol{Y},\boldsymbol{Y}' \overset{\text{i.i.d.}}{\sim} F$*. The two* $\mathbb{E}$ *symbols represent the expectation operators applied to* $\boldsymbol{Y}$ *and* $\boldsymbol{Y},\boldsymbol{Y}'$*, respectively.*

Proof of Proposition 3.3. Since $\{\boldsymbol{x}_i\}_{i=1}^N$ are N samples of F, and N is usually large, for any $\theta \in \Theta$, $\int_{\mathcal{X}} \log f(\boldsymbol{x}|\theta)dF$ and $E(F, F_{U_n})$ can be approximated by their Monte Carlo approximations $\frac{1}{N}\sum_{i=1}^N \log f(\boldsymbol{x}_i|\theta)$ and $ED(F_N, F_{U_n})$, respectively. Hence, according to Lemma A.1, we can directly obtain Proposition 3.3. □

We now preface the proof Theorem 3.4 with three lemmas, some of which are inspired by Medina-Aguayo et al. [16]. Recall that the (standard) MH acceptance ratio is $\alpha(\theta,\theta') = 1 \wedge a(\theta,\theta')$. For notational simplicity, define $\phi_{U_n^*}(\theta) = \prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta)^{N/n} / \prod_{i=1}^N f(\boldsymbol{x}_i \mid \theta)$.

Lemma A.2. *For any* $(\theta,\theta') \in \Theta^2$*, we have*

$$\alpha_{U_n^*}(\theta,\theta') \leqslant \alpha(\theta,\theta')\left(1 \vee \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right).$$

Proof. Recall that the definitions of $\alpha_{U_n^*}(\theta,\theta')$ and $a_{U_n^*}(\theta,\theta')$ in (3.3), we have

$$\begin{aligned}
\alpha_{U_n^*}(\theta,\theta') &= 1 \wedge a_{U_n^*}(\theta,\theta')\\
&= 1 \wedge \frac{\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta')^{N/n} p(\theta') q(\theta|\theta') \prod_{i=1}^N f(\boldsymbol{x}_i|\theta) \prod_{i=1}^N f(\boldsymbol{x}_i|\theta')}{\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta)^{N/n} p(\theta) q(\theta'|\theta) \prod_{i=1}^N f(\boldsymbol{x}_i|\theta') \prod_{i=1}^N f(\boldsymbol{x}_i|\theta)}\\
&= 1 \wedge \left(a(\theta,\theta') \frac{\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta')^{N/n}}{\prod_{i=1}^N f(\boldsymbol{x}_i|\theta')} \frac{\prod_{i=1}^N f(\boldsymbol{x}_i|\theta)}{\prod_{j=1}^n f(\boldsymbol{x}_j^{U_n^*}|\theta)^{N/n}}\right)\\
&= 1 \wedge \left(a(\theta,\theta') \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right)\\
&\leqslant 1 \wedge \left[a(\theta,\theta')\left(1 \vee \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right)\right]\\
&\leqslant (1 \wedge a(\theta,\theta'))\left(1 \vee \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right)\\
&= \alpha(\theta,\theta')\left(1 \vee \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right),
\end{aligned}$$

where the second "$\leqslant$" sign holds following from the fact that the inequality $1 \wedge ab \leqslant (1 \wedge a)b$ holds for $a > 0$ and $b \geqslant 1$, and $a \vee b = \max\{a, b\}$. □

Lemma A.3. *For any* $(\theta,\theta') \in \Theta^2$*, let* $\tau_{\max} = \sup_{\theta\in\Theta} \tau(\theta)$*, we have*

$$\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)} \leqslant \mathrm{e}^{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}}. \tag{A.1}$$

Proof. It holds that

$$\begin{aligned}
\log \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)} &= \log \phi_{U_n^*}(\theta') - \log \phi_{U_n^*}(\theta)\\
&= N\left(\frac{1}{n}\sum_{j=1}^n \log f(\boldsymbol{x}_j^{U_n^*}|\theta') - \frac{1}{N}\sum_{i=1}^N \log f(\boldsymbol{x}_i|\theta')\right)\\
&\quad + N\left(\frac{1}{N}\sum_{i=1}^N \log f(\boldsymbol{x}_i|\theta) - \frac{1}{n}\sum_{j=1}^n \log f(\boldsymbol{x}_j^{U_n^*}|\theta)\right)\\
&\leqslant N\left|\frac{1}{n}\sum_{j=1}^n \log f(\boldsymbol{x}_j^{U_n^*}|\theta') - \frac{1}{N}\sum_{i=1}^N \log f(\boldsymbol{x}_i|\theta')\right|
\end{aligned}$$

$$
\begin{aligned}
&+ N\bigg|\frac{1}{N}\sum_{i=1}^{N}\log f(\boldsymbol{x}_i|\theta) - \frac{1}{n}\sum_{j=1}^{n}\log f(\boldsymbol{x}_j^{U_n^*}|\theta)\bigg| \\
&\leqslant 2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}.
\end{aligned}
$$

The last "$\leqslant$" holds following from Lemma A.1. Hence, we obtain (A.1). □

Lemma A.4. *Recall that* $\rho(\theta) = 1 - \int_\Theta \alpha(\theta,\theta')q(\theta'|\theta)d\theta'$ *and* $\rho_{U_n^*}(\theta) = 1 - \int_\Theta a_{U_n^*}(\theta,\theta')q(\theta'|\theta)d\theta'$. *Then for any* $\theta \in \Theta$, *we have*

$$\rho_{U_n^*}(\theta) - \rho(\theta) \leqslant \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg|\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)} - 1\bigg|d\theta'.$$

Proof. According to the inequality $1 \wedge ab \geqslant (1 \wedge a)(1 \wedge b)$ for $a, b \geqslant 0$, and the proof of Lemma A.2, we have

$$\alpha_{U_n^*}(\theta,\theta') = 1 \wedge \bigg(a(\theta,\theta')\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg) \geqslant (1 \wedge a(\theta,\theta'))\bigg(1 \wedge \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg) = \alpha(\theta,\theta')\bigg(1 \wedge \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg).$$

So

$$\rho_{U_n^*}(\theta) = 1 - \int_\Theta q(\theta'|\theta)\alpha_{U_n^*}(\theta,\theta')d\theta' \leqslant 1 - \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg(1 \wedge \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg)d\theta'.$$

Hence,

$$
\begin{aligned}
\rho_{U_n^*}(\theta) - \rho(\theta) &\leqslant \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')d\theta' - \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg(1 \wedge \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg)d\theta' \\
&= \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg[0 \vee \bigg(1 - \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg)\bigg]d\theta' \\
&\leqslant \int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg|\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)} - 1\bigg|d\theta'.
\end{aligned}
$$

This completes the proof. □

Proof of Theorem 3.4. We first establish the equivalence between geometric ergodicity and a geometric drift condition. For any kernel K, let

$$KV(\boldsymbol{x}) = \int_\Theta K(\boldsymbol{x},\boldsymbol{z})V(\boldsymbol{z})d\boldsymbol{z}.$$

From [18, Theorems 14.0.1 and 15.0.1], there exist a function $V : \Theta \to [1,\infty]$, two constants $\lambda \in (0,1)$ and $b < \infty$ and a small set $S \subset \Theta$ such that K satisfies a drift condition, i.e.,

$$KV \leqslant \lambda V + b\mathbf{1}_S. \tag{A.2}$$

We now show how to use the previous lemmas to establish the geometric ergodicity of $K_{U_n^*}$ for some sufficiently large n. The proof is very similar to that presented in [16, Theorem 3.2]. It holds that

$$
\begin{aligned}
(K_{U_n^*} - K)V(\theta) &= \int_\Theta q(\theta'|\theta)(\alpha_{U_n^*}(\theta,\theta') - \alpha(\theta,\theta'))V(\theta')d\theta' + (\rho_{U_n^*}(\theta) - \rho(\theta))V(\theta) \\
&\leqslant \int_\Theta \bigg[\bigg(1 \vee \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\bigg) - 1\bigg]q(\theta'|\theta)\alpha(\theta,\theta')V(\theta')d\theta' \\
&\quad + \bigg(\int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')\bigg|\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)} - 1\bigg|d\theta'\bigg)V(\theta) \\
&\leqslant (\exp\{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}\} - 1)\int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')V(\theta')d\theta'
\end{aligned}
$$

$$
\begin{aligned}
&+ (\exp\{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}\} - 1)V(\theta)\int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')d\theta' \\
&\leqslant (\exp\{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}\} - 1)(\lambda V(\theta) + b\mathbf{1}_S(\theta) - \rho(\theta)V(\theta)) \\
&+ (\exp\{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}\} - 1)\left(V(\theta)\int_\Theta q(\theta'|\theta)\alpha(\theta,\theta')d\theta'\right) \\
&\leqslant (C_{N,U_n^*} - 1)b\mathbf{1}_S(\theta) + (C_{N,U_n^*} - 1)(\lambda + 1 - 2\rho(\theta))V(\theta),
\end{aligned}
\tag{A.3}
$$

where $C_{N,U_n^*} = \exp\{2N\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})}\}$. Fix $\epsilon > 0$. According to the definition of $ED(F_N, F_{U_n^*})$, there exists an $n_0 < N$ such that

$$
n \geqslant n_0, \quad C_{N,U_n^*} - 1 \leqslant \epsilon.
$$

Combining (A.2) and (A.3), we have

$$
K_{U_n^*}V(\theta) \leqslant \epsilon b\mathbf{1}_S(\theta) + \epsilon(\lambda + 1 - 2\rho(\theta))V(\theta).
$$

Based on the condition $\sup_{\theta\in\Theta}\rho(\theta) < 1/2$, if we take $\epsilon < 1/(1 + \lambda - 2\rho(\theta))$, we can show that $K_{U_n^*}$ (for $n \geqslant n_0$) satisfies a geometric drift condition. Then according to [16, Theorem 3.2], $K_{U_n^*}$ is geometrically ergodic. □

Proof of Theorem 3.5. This proof borrows ideas from the perturbation analysis of uniformly ergodic Markov chains. First, by straightforward algebra, we have

$$
\begin{aligned}
\|K(\theta,\cdot) - K_{U_n^*}(\theta,\cdot)\|_{TV} &\leqslant \int_\Theta q(\theta'|\theta)|\alpha(\theta,\theta') - \alpha_{U_n^*}(\theta,\theta')|d\theta' \\
&\leqslant \int_\Theta q(\theta'|\theta)|a(\theta,\theta') - a_{U_n^*}(\theta,\theta')|d\theta' \\
&= \int_\Theta q(\theta'|\theta)a(\theta,\theta')\left|1 - \frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right|d\theta' \\
&= \int_\Theta q(\theta'|\theta)a(\theta,\theta')\frac{|\phi_{U_n^*}(\theta') - \phi_{U_n^*}(\theta)|}{\phi_{U_n^*}(\theta)}d\theta' \\
&\leqslant A_n \sup_{\theta\in\Theta} B_n(\theta).
\end{aligned}
\tag{A.4}
$$

Now, under the assumption of uniform ergodicity, i.e., $\sup_{\theta\in\Theta}\|\pi - K^i(\theta,\cdot)\|_{TV} \leqslant C\varrho^i$, using [19, Corollary 3.1], we have that for any starting point $\theta \in \Theta$,

$$
\|K^i(\theta,\cdot) - K^i_{U_n^*}(\theta,\cdot)\|_{TV} \leqslant \left(\lambda + \frac{C\varrho^\lambda}{1-\varrho}\right)\sup_{\theta\in\Theta}\|K(\theta,\cdot) - K_{U_n^*}(\theta,\cdot)\|_{TV},
\tag{A.5}
$$

where $\lambda = \lceil\log(1/C)\log\varrho\rceil$. Combining (A.4) and (A.5) leads to (3.4) with $\kappa = \lambda + C\varrho^\lambda/1 - \varrho$. Moreover, using (3.4) and (A.5), we have

$$
\begin{aligned}
\sup_{\theta\in\Theta}\|\pi - K^i_{U_n^*}(\theta,\cdot)\|_{TV} &\leqslant \sup_{\theta\in\Theta}\|\pi - K^i(\theta,\cdot)\|_{TV} + \sup_{\theta\in\Theta}\|K^i(\theta,\cdot) - K^i_{U_n^*}(\theta,\cdot)\|_{TV} \\
&\leqslant C\varrho^i + \kappa A_n \sup_{\theta\in\Theta} B_n(\theta),
\end{aligned}
$$

and taking the limit when $i \to \infty$ leads to (3.5). Finally, for a large enough n, we know from Theorem 3.4 that the Markov chain produced by EDSS-MCMC is geometrically ergodic. For such an n, we have that for any $\theta \in \Theta$,

$$
\begin{aligned}
\|\pi - \pi_n^*\|_{TV} &\leqslant \|K^i(\theta,\cdot) - \pi\|_{TV} + \|K^i_{U_n^*}(\theta,\cdot) - \pi_n^*\|_{TV} + \|K^i(\theta,\cdot) - K^i_{U_n^*}(\theta,\cdot)\|_{TV} \\
&\leqslant \|K^i(\theta,\cdot) - \pi\|_{TV} + \|K^i_{U_n^*}(\theta,\cdot) - \pi_n^*\|_{TV} + \kappa A_n \sup B_n(\theta),
\end{aligned}
$$

and taking the limit $i \to \infty$ yields (3.6). □

Proof of Corollary 3.6. It holds that

$$\log \phi_{U_n^*}(\theta) = N\left(\frac{1}{n}\sum_{j=1}^{n}\log f(\boldsymbol{x}_j^{U_n^*}|\theta) - \frac{1}{N}\sum_{i=1}^{N}\log f(\boldsymbol{x}_i|\theta)\right)$$

$$\leqslant N\left|\frac{1}{N}\sum_{i=1}^{N}\log f(\boldsymbol{x}_i|\theta) - \frac{1}{n}\sum_{j=1}^{n}\log f(\boldsymbol{x}_j^{U_n^*}|\theta)\right|.$$

According to Lemma A.1, we have

$$\frac{1}{\phi_{U_n^*}(\theta)} \leqslant \exp\{N\tau(\theta)\sqrt{ED(F_N, F_{U_n^*})}\} \quad \text{and} \quad \phi_{U_n^*}(\theta) \leqslant \exp\{N\tau(\theta)\sqrt{ED(F_N, F_{U_n^*})}\}.$$

Then $A_n(\theta)$ and $B_n(\theta)$ in Theorem 3.5 satisfy

$$A_n \leqslant \sup_{\theta\in\Theta}\exp\{N\tau(\theta)\sqrt{ED(F_N, F_{U_n^*})}\}$$

and

$$B_n(\theta) \leqslant 2\sup_{\theta\in\Theta}\exp\{N\tau(\theta)\sqrt{ED(F_N, F_{U_n^*})}\}D_n(\theta).$$

Hence,

$$A_n\sup_{\theta\in\Theta}B_n(\theta) \leqslant 2\sup_{\theta\in\Theta}\exp\{2N\tau(\theta)\sqrt{ED(F_N, F_{U_n^*})}\}D_n(\theta).$$

Submitting $A_n\sup_{\theta\in\Theta}B_n(\theta)$ in Theorem 3.5, we can obtain Corollary 3.6. □

Proof of Remark 3.7. For the uniform (Uni) subsampling probabilities method,

$$\Lambda_n^{\mathrm{Uni}}(\theta,\theta') = l_n^{\mathrm{Uni}}(\theta') - l_n^{\mathrm{Uni}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log f(\boldsymbol{x}_i^*|\theta') - \frac{1}{n}\sum_{i=1}^{n}\log f(\boldsymbol{x}_i^*|\theta),$$

where $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_n^*$ represent n random samples taken from the full data with replacement, i.e.,

$$P(\boldsymbol{x}_i^* = \boldsymbol{x}_j) = 1/N,$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, N$. Hence, we have

$$E[|\Lambda_n^{\mathrm{Uni}}(\theta,\theta') - \Lambda(\theta,\theta')|] = E\left[\left|\frac{1}{n}\sum_{i=1}^{n}\log\frac{f(\boldsymbol{x}_i^*|\theta')}{f(\boldsymbol{x}_i^*|\theta)} - \Lambda(\theta,\theta')\right|\right]$$

$$\leqslant \frac{1}{n}\sum_{i=1}^{n}E\left|\log\frac{f(\boldsymbol{x}_i^*|\theta')}{f(\boldsymbol{x}_i^*|\theta)} - \Lambda(\theta,\theta')\right|$$

$$= \frac{1}{N}\sum_{j=1}^{N}\left|\log\frac{f(\boldsymbol{x}_j|\theta')}{f(\boldsymbol{x}_j|\theta)} - \Lambda(\theta,\theta')\right|.$$

For the MLO subsampling probabilities method,

$$\Lambda_n^{\mathrm{MLO}}(\theta,\theta') = l_n^{\mathrm{MLO}}(\theta') - l_n^{\mathrm{MLO}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta') - \frac{1}{n}\sum_{i=1}^{n}\log f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta),$$

where $\{\boldsymbol{x}_1^{\mathrm{MLO}}, \ldots, \boldsymbol{x}_n^{\mathrm{MLO}}\}$ are obtained by the nonuniform subsampling probabilities

$$\boldsymbol{\eta}^{\mathrm{opt}} = (\eta_1^{\mathrm{opt}}, \ldots, \eta_N^{\mathrm{opt}})$$

with replacement, i.e., $P(\boldsymbol{x}_i^{\mathrm{MLO}} = \boldsymbol{x}_j) = \eta_j^{\mathrm{opt}}$. Hence, we have

$$E[|\Lambda_n^{\mathrm{MLO}}(\theta,\theta') - \Lambda(\theta,\theta')|] = E\left[\left|\frac{1}{n}\sum_{i=1}^{n}\frac{1}{N\eta_i^{\mathrm{opt}}}\log\frac{f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta')}{f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta)} - \Lambda(\theta,\theta')\right|\right]$$

$$
\begin{aligned}
&\leqslant \frac{1}{n}\sum_{i=1}^{n} E\bigg|\frac{1}{N\eta_i^{\mathrm{opt}}}\log\frac{f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta')}{f(\boldsymbol{x}_i^{\mathrm{MLO}}|\theta)} - \Lambda(\theta,\theta')\bigg| \\
&= \sum_{j=1}^{N}\bigg|\frac{1}{N}\log\frac{f(\boldsymbol{x}_j|\theta')}{f(\boldsymbol{x}_j|\theta)} - \eta_j^{\mathrm{opt}}\Lambda(\theta,\theta')\bigg|.
\end{aligned}
$$

For the energy distance subsampling (EDSS) method,

$$
\Lambda_n^{\mathrm{EDSS}}(\theta,\theta') = \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\boldsymbol{x}_i^{U_n^*}|\theta')}{f(\boldsymbol{x}_i^{U_n^*}|\theta)}\right].
$$

Hence,

$$
\begin{aligned}
|\Lambda_n^{\mathrm{EDSS}}(\theta,\theta') - \Lambda(\theta,\theta')| &= \left|\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\boldsymbol{x}_i^{U_n^*}|\theta')}{f(\boldsymbol{x}_i^{U_n^*}|\theta)}\right] - \frac{1}{N}\sum_{j=1}^{N}\log\frac{f(\boldsymbol{x}_j|\theta')}{f(\boldsymbol{x}_j|\theta)}\right| \\
&= \left|\frac{1}{N}\log\frac{\phi_{U_n^*}(\theta')}{\phi_{U_n^*}(\theta)}\right| \\
&\leqslant 2\tau_{\max}\sqrt{ED(F_N, F_{U_n^*})},
\end{aligned}
$$

where the last inequality is obtained based on Lemma A.3. □