Robust Gradient-Based Markov Subsampling

Tieliang Gong, Quanhan Xi, Chen Xu
Deparment of Mathematics and Statistics, University of Ottawa, Ottawa, ON, K1N6N5, Canada

Abstract

Subsampling is a widely used and effective method to deal with the challenges brought by big data. Most subsampling procedures are designed based on the importance sampling framework, where samples with high importance measures are given corresponding sampling probabilities. However, in the highly noisy case, these samples may cause an unstable estimator which could lead to a misleading result. To tackle this issue, we propose a gradient-based Markov subsampling (GMS) algorithm to achieve robust estimation. The core idea is to construct a subset which allows us to conservatively correct a crude initial estimate towards the true signal. Specifically, GMS selects samples with small gradients via a probabilistic procedure, constructing a subset that is likely to exclude noisy samples and provide a safe improvement over the initial estimate. We show that the GMS estimator is statistically consistent at a rate which matches the optimal in the minimax sense. The promising performance of GMS is supported by simulation studies and real data examples.

Introduction

The rapid development of science and technology in the past decade have introduced data of extraordinary size and complexity, which brings great challenges to conventional machine learning and statistical methods. A popular way to deal with big data is the divide and conquer strategy, which involves partitioning the data, running a particular learning algorithm on data segments in parallel, and then aggregating a global output by combining these individual parallel outputs. To this end, distributed computing platforms like Spark (Zaharia et al. 2010) and Ray (Moritz and et al 2018) have been developed. As a computationally cheaper alternative, subsampling techniques have also drawn a great deal of attention for processing big data (Fithian and Hastie 2014; Halko, Martinsson, and Tropp 2011; Dereziński, Warmuth, and Hsu 2018a; Derezinski, Warmuth, and Hsu 2018b). These methods aim to select a representative subset from the full data for downstreaming learning tasks. The computational burden is then greatly alleviated as the selected subset is quite small.

Intuitively, uniform sampling is perhaps the most straightforward way to conduct subsampling. However, uniform sampling can be inefficient and unstable for data with high noise level. Therefore, we usually resort to informative subsampling, where important observations are given a higher chance to be selected. One representative informative subsampling procedure is leverage score sampling (Drineas et al. 2011; 2012; Drineas, Mahoney, and Muthukrishnan 2008; Rudi et al. 2018), which assigns sampling probability proportional to a distance measure within the covariates. Another informative subsampling procedure is given in (Zhu 2016), where the sampling probabilities are computed proportional to the quadratic loss gradient using a pilot estimator. Gradient-based subsampling (GS) notably uses information derived from the input data as well as the response. Following in this spirit, the work in (Ting and Brochu 2018) proposes an influence function as a information measure to calculate sampling probabilities, which is shown to lead to an asymptotically optimal sampling distribution in the sense of minimum variance for the resulting linear estimator.

These informative subsampling methods usually boil down to solving a weighted least squares problem which is sensitive to unbalanced sampling probabilities. Consequently, the resulting estimators can be less precise in applications (Ma, Mahoney, and Yu 2015). Moreover, these methods assign higher probabilities to samples that are highly influential to the estimator. However, many of these samples can be noisy or outliers when the noise level is high, and these estimators would result in a misleading conclusion. In addition, the accuracy of GS and influence sampling depends on a reliable initial estimator which would be difficult to obtain in the noisy setting. In this paper, we develop a robust gradient-based Markov subsampling (GMS) method for linear regression. The procedure is as follows: We first obtain a crude estimator $\boldsymbol{\beta}_0$ based on a simple pilot selection. We later show that by selecting samples with small loss gradients, it is possible to construct a subset $\mathcal{D}_S$ on which the empirical loss at $\boldsymbol{\beta}_0$ serves as a good approximation of the empirical loss at the oracle $\boldsymbol{\beta}^*$. Since $\boldsymbol{\beta}_0$ is a rough estimator to $\boldsymbol{\beta}^*$ and quadratic loss is convex, the least square estimator $\hat{\boldsymbol{\beta}}$ on $\mathcal{D}_S$ is a safe improvement from $\boldsymbol{\beta}_0$ towards $\boldsymbol{\beta}^*$ (See Fig. 1 for an illustration). The pro-

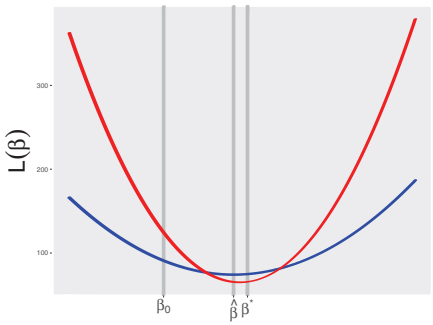


Figure 1: A random sample $n = 200$ is generated from $y_i = \mathbf{x}_i\boldsymbol{\beta}_i^* + \varepsilon_i$ where $\mathbf{x}_i \sim \mathbf{N}(0, 4)$, $\varepsilon_i \sim \mathbf{U}(-2, 2)$. The red line denotes the quadratic loss on full data, and the blue line denotes the loss function on a subset $\mathcal{D}_S$ generated by the GMS procedure.

posed GMS constructs such a $\mathcal{D}_S$ by selecting samples of small gradient value through a Metropolis-Hastings (MH) type procedure. By doing so, samples with large gradient value are unlikely to be selected and outliers are avoided with high probability. We show that under mild conditions, the GMS estimator is statistically consistent with a rate at order $\mathcal{O}(\sqrt{d\log(d)/n_{sub}})$, where n_{sub} denotes the subsample size and d is the number of regression covariates. The superior performance of GMS is also supported by simulation studies and real-world examples.

The rest of this paper is organized as follows: Section 2 sets the notations and problem statement. Section 3 introduces the proposed GMS algorithm. Section 4 establishes the corresponding error bound. Section 5 presents experimental results on both simulations and real-world dataset. Section 6 concludes our work.

Notations and Preliminaries

To make our arguments in the following sections precise, some concepts and notations are introduced.

Definition 1 *(Vershynin 2018) A random variable $X \in \mathbb{R}$ is said to be sub-Gaussian with variance proxy σ^2 if $\mathbb{E}[X] = 0$ and its moment generating function satisfies*

$$\mathbb{E}[\exp(sX)] \leq \exp\left(\frac{s^2\sigma^2}{2}\right), \ \forall s \in \mathbb{R}.$$

We denote a sub-Gaussian random variable as $X \sim \mathrm{subG}(\sigma^2)$. More generally, a random vector $X \in \mathbb{R}^d$ is said to be sub-Gaussian with variance proxy σ^2 if $\mathbb{E}[X] = 0$ and $\mathbf{u}^\top X$ is sub-Gaussian with variance proxy σ^2 for any unit vector $\mathbf{u} \in \mathcal{S}^{d-1}$. In this case we write $X \sim \mathrm{subG}_d(\sigma^2)$.

In this paper, we consider a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ generated according to the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{n\times d}$ is a design matrix, $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top \in \mathbb{R}^n$ is a response vector , $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots \varepsilon_n)^\top$ and $\varepsilon_i \sim \mathrm{subG}(\sigma^2)$ for $i = 1, 2, \cdots, n$. Denote the quadratic loss on $\mathcal{D}$ by $L(\boldsymbol{\beta}) = \sum_{i=1}^n \ell_i(\boldsymbol{\beta})$, where $\ell_i(\boldsymbol{\beta}) = (y_i - \mathbf{x}_i\boldsymbol{\beta})^2$.

We focus on the setting $n \gg d$, where the least squares solution to model (1) is given by

$$\boldsymbol{\beta}_n = \frac{1}{n}\Sigma_n^{-1}\mathbf{X}^\top\mathbf{y}, \tag{2}$$

where $\Sigma_n = \mathbf{X}^\top\mathbf{X}/n$ denotes the empirical covariance matrix. Classic algorithms, including Cholesky decomposition, QR decomposition and Singular Value Decomposition compute $\boldsymbol{\beta}_n$ in $\mathcal{O}(nd^2)$ time. For a matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$, we denote its maximal and minimal eigenvalues by $\lambda_{\max}(\mathbf{M})$ and $\lambda_{\min}(\mathbf{M})$. For a vector $\mathbf{u} \in \mathbb{R}^d$, we denote its ℓ_2 norm by $\|\mathbf{u}\|$.

The following concepts play an important role in our theoretical analysis. Let $\{X_i\}_{i\geq 1}$ be a Markov chain on a general space $\mathcal{X}$ with invariant probability distribution π. Let $P(x, \mathrm{d}y)$ be a Markov transition kernel on a general space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ and P^* be its adjoint. Denote $L_2(\pi)$ by the Hilbert space consisting of square integrable functions with respect to π. For any function $h : \mathcal{X} \to \mathbb{R}$, we write $\pi(h) := \int h(x)\pi(\mathrm{d}x)$. Denote the additive reversiblization by $R = (P + P^*)/2$. Let $P^t(x, \mathrm{d}y), (t \in \mathbb{N})$ be the t-step Markov transition kernel corresponding to P, then for $i \in \mathbb{N}, x \in \mathcal{X}$ and a measurable set S, $P^t(x, S) = \Pr(X_{t+i} \in S|X_i = x)$. Following the above notations, we introduce the definitions of ergodicity and spectral gap for a Markov chain.

Definition 2 *Let $M(x)$ be a non-negative function. For an initial probability measure $\rho(\cdot)$ on $\mathcal{B}(\mathcal{X})$, a Markov chain is uniformly ergodic if*

$$\|P^t(\rho, \cdot) - \pi(\cdot)\|_{TV} \leq M(x)t^n \tag{3}$$

for some $M(x) < \infty$ and $t < 1$, where $\|\cdot\|_{TV}$ denotes total variation norm.

A Markov chain is geometrically ergodic if (3) holds for some $t < 1$, which eliminates the bounded assumption on $M(x)$.

Definition 3 *(Absolute spectral gap) A Markov operator P admits an absolute spectral gap $1 - \lambda$ if*

$$\lambda := \sup\{\|Ph\|_\pi : \|h\|_\pi = 1, \pi(h) = 0\} < 1.$$

Let $\alpha(\lambda) := \frac{1+\lambda}{1-\lambda}$. Obviously, $\alpha(\lambda)$ is strictly increasing with λ and $\alpha(\lambda) = 1$ as $\lambda = 0$.

Definition 4 *(Right spectral gap) A Markov operator P admits an right spectral gap $1 - \lambda_r$ if R has $\lambda_r < 1$, where*

$$\lambda_r := \sup\{\langle Rh, h\rangle : \|h\|_\pi = 1, \pi(h) = 0\}.$$

The spectral gap measures the convergence rate of a Markov chain towards its invariant state π (Rudolf 2011). The bigger the spectral gap, the faster the convergence to the stationary distribution. Since R is self-adjoint, it is known that $\lambda_r \leq \lambda$.

Robust Gradient-based Markov Subsampling

The idea of gradient-based learning has been studied in the literature (Zhu 2016; Burke, Lewis, and Overton 2005; Burke et al. 2018). Recall that GS selects samples with probability directly proportional to their gradient values, providing a natural way to apply this idea to subsampling. Although GS performs well empirically, the resulting estimator can be sensitive to highly noisy data, particularly when the sampling ratio is small. In contrast, we explore the potential of selecting samples with small gradients to achieve robust estimation. To this end, we develop a gradient-based Markov subsampling (GMS) algorithm. Concretely, GMS consists of three steps: 1) pilot estimation; 2) gradient calculation; 3) Markov subsampling.

- **Pilot estimation.** The pilot estimation β_0 can be calculated by (2) based on a small random subset with size $n_0 \ll n$. We empirically show that the GMS does not heavily rely on the quality of $\boldsymbol{\beta}_0$. As a result, n_0 can be chosen to the user's preference in practice.
- **Calculate gradient.** Given the pilot $\boldsymbol{\beta}_0$, we calculate the gradient for the i-th sample by

$$\mathbf{g}_i(\boldsymbol{\beta}_0) = \frac{\partial \ell_i(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}|_{\boldsymbol{\beta}=\boldsymbol{\beta}_0} = -\mathbf{x}_i^\top(y_i - \mathbf{x}_i\boldsymbol{\beta}_0). \quad (4)$$

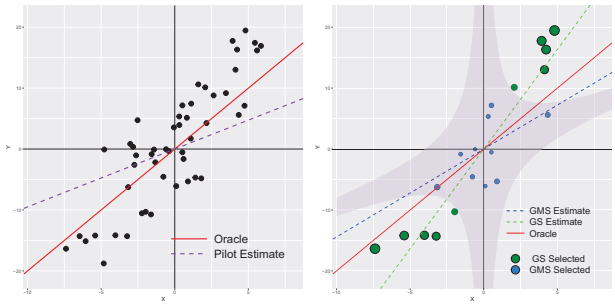
  As discussed previously, we are attempting to find a subset $\mathcal{D}_S$ on which $L(\boldsymbol{\beta}^*) \approx L(\boldsymbol{\beta}_0)$. Consider the first order Taylor expansion of $L(\boldsymbol{\beta}^*)$ centered at the pilot $\boldsymbol{\beta}_0$:

$$L(\boldsymbol{\beta}^*) = L(\boldsymbol{\beta}_0) + \sum_{i=1}^{n} \langle \mathbf{g}_i(\boldsymbol{\beta}_0), \boldsymbol{\beta}_0 - \boldsymbol{\beta}^* \rangle + \mathcal{R}_n(\boldsymbol{\beta}_0),$$

  where $\mathcal{R}_n$ is the remainder. Since $\boldsymbol{\beta}_0$ is considered to be a crude estimate for $\boldsymbol{\beta}^*$, $\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*$ is non-negligible. Thus, a natural way to satisfies $L(\boldsymbol{\beta}_0) \approx L(\boldsymbol{\beta}^*)$ is to select points with small $\mathbf{g}_i(\boldsymbol{\beta}_0), i \in \mathcal{D}_S$.
- **Markov subsampling.** It has been empirically shown that Markov chain samples may lead to more stable estimation compared to their i.i.d. counterparts (Gong, Zou, and Xu 2015; Sun, Sun, and Yin 2018). Taking this in mind, we implement probabilistic sampling through a Metropolis-Hastings (MH) type procedure. Since we prefer samples with small gradient values, we use these values to design the probability in our acceptance step. This procedure can be summarized as follows. At some current sample $\mathbf{z}_t$, we accept a randomly selected candidate sample $\mathbf{z}^*$ with probability defined in (5). If accepted, the iteration is completed and we set $\mathbf{z}^* = \mathbf{z}_{t+1}$. Otherwise, a new candidate is randomly selected and the process repeats. Finally, we accept the last n_0 elements generated by this procedure after a user-specified burn-in period t_0.

The detailed procedure is summarized in Algorithm 1.

Remark 1 *It is known that the the subsamples generated by Algorithm 1 constitute an irreducible Markov Chain (since the transition probabilities are always positive), and therefore are uniformly ergodic (Down, Meyn, and Tweedie 1995).*



(a) Full data (b) Subsampled data

Figure 2: (a) Oracle and pilot estimation; (b) Comparison on 10 data points selected by GS and GMS. The size of the subsampled data points are scaled by their corresponding gradient value.

Remark 2 *The overall computational complexity of GMS is* $\mathcal{O}(\max\{n_{sub}d^2, d^3\})$. *Thus, the computational burden can be greatly reduced when* $n_{sub} \ll n$.

We give a toy example to show the potential benefits of GMS. We generate a dataset of size 50 by $y_i = 2\mathbf{x}_i + \varepsilon_i$, where $\mathbf{x}_i \sim \mathbf{N}(0, 3)$, $\varepsilon_i \sim \mathbf{U}(-10, 10)$. Fig. 2 (a) plots the oracle (red solid line) and pilot estimation (purple dashed line) which is fitted by 10 randomly selected data points. Fig. 2(b) demonstrates the comparison of 10 samples selected by GS (green circles), GMS (blue circles) respectively. The purple area denotes the range where the gradient value is less than 15. It can be observed from Fig. 2 that samples in purple area help to correct the pilot line towards the oracle. In other words, by selecting the data points with small gradient values, GMS can safely improve the pilot estimation while also avoiding distractions caused by noisy samples.

Algorithm 1 Robust Gradient-based Markov Subsampling

Input: Dataset $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^n$, $\mathcal{D}_S = \emptyset$, burn-in period: t_0, subsample size $n_{sub} \ll n$.

1: Train a pilot estimator β_0 based on a subsample with size $n_0 = n_{sub}$ and calculate $\mathbf{g}_i, i = 1, 2, \cdots, n$ by (4).
2: Randomly select a sample $\mathbf{z}_1$ from $\mathcal{D}$, and set $\mathcal{D}_s = \mathbf{z}_1$.
3: **for** $2 \le t \le (n_{sub} + t_0)$ **do**
4: **while** $|\mathcal{D}_S| < t$ **do**
5: Randomly draw a candidate $\mathbf{z}^* = (\mathbf{x}^*, y^*)$
6: Calculate the acceptance probability by

$$p = \min\left\{1, \frac{\|\mathbf{g}_t\|}{\|\mathbf{g}^*\|}\right\} \quad (5)$$

7: Set $\mathcal{D}_S = \mathcal{D}_S \cup \mathbf{z}^*$ with probability p
8: If $\mathbf{z}^*$ is accepted, set $\mathbf{z}_{t+1} = \mathbf{z}^*$
9: **end while**
10: **end for**
11: Denote the last n_{sub} samples of $\mathcal{D}_S$ as $(\mathbf{X}_S, \mathbf{y}_S)$.
12: Solve $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y}_S - \mathbf{X}_S\boldsymbol{\beta}\|^2$.

Output: $\hat{\boldsymbol{\beta}}$.

Theoretical Assessment

In this section, we provide theoretical support for the proposed GMS. In particular, our main interest is to bound the difference between the GMS estimator $\hat{\boldsymbol{\beta}}$ and the oracle $\boldsymbol{\beta}^*$. The main result is given in Theorem 1, which shows that the gradient based Markov subsampling algorithm is statistically consistent. We first present several lemmas as follows.

Lemma 1 *(Vershynin 2018) If $X \sim \mathrm{subG}(\sigma^2)$, then for any $t > 0$, it holds $\mathbb{P}(|X| \geq t) \leq 2\exp\left(\frac{-t^2}{2\sigma^2}\right)$.*

Lemma 2 *(Sun et al. 2017) Let $A, B \in \mathbb{R}^{d\times d}$ be invertible, then for any matrix norm $\|\cdot\|_*$, if $\|A^{-1}\|_*\|A-B\|_* < 1$, we have*

$$\|A^{-1} - B^{-1}\|_* \leq \frac{\|A^{-1}\|_*^2\|A-B\|_*}{1-\|A^{-1}\|_*\|A-B\|_*}.$$

Lemma 3 *(Fan, Jiang, and Sun 2018) Let $\{\mathbf{x}_i\}_{i\geq 1}$ be a Markov chain with invariant measure π and right spectral gap $1-\lambda_r > 0$. Then for any bounded function $f : \mathcal{X} \to [a,b]$ and any $t \in \mathbb{R}$,*

$$\mathbb{P}\left(\left|\sum_{i=1}^n f(\mathbf{x}_i) - n\pi(f)\right| > \epsilon\right) \leq 2\exp\left(-\frac{\alpha(\lambda_r \vee 0)^{-1}\epsilon^2}{2n(b-a)^2/4}\right).$$

Theorem 1 *Suppose that the Markov chain samples generated by GMS are with invariant distribution π and right spectral gap $1-\lambda_r$. Let $\hat{\Sigma}_S = \frac{1}{n_{sub}}\mathbf{X}_S^\top\mathbf{X}_S, \Sigma = cov(\mathbf{X})$ if $n_{sub} \geq 8\alpha(\lambda_r \vee 0)d^2\|\Sigma^{-1}\|^2\log(d^2/\delta)$, then with confidence at least $1-2\delta$,*

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| \leq C\sqrt{\frac{8d(\log(d)+\log(1/\delta))}{n_{sub}}}, \qquad (6)$$

where $C = \sigma\|\Sigma^{-1}\|^2$.

PROOF. Note that the Markov chain samples $\{\mathbf{x}_i\}_{i=1}^{n_{sub}}$ are uniformly ergodic. Suppose that it has a stationary invariant distribution π with right spectral gap $1-\lambda_r$. Without loss of generality, we assume that $\sup_{\mathbf{x}_j\in\mathcal{X}}\|\mathbf{x}_j\| \leq 1$ for $j = 1, 2, \cdots, n_{sub}$. Observe that

$$\begin{aligned}
&\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| \\
&= \|(n_{sub}^{-1}\mathbf{X}_S^\top\mathbf{X}_S)^{-1}(n_{sub}^{-1}\mathbf{X}_S^\top\mathbf{y}_S) - \boldsymbol{\beta}^*\| \\
&= \|(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{X}_S^\top\mathbf{X}_S\boldsymbol{\beta}^* - \boldsymbol{\beta}^* + (n_{sub}^{-1}\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\frac{\mathbf{X}_S^\top\boldsymbol{\varepsilon}}{n_{sub}}\| \\
&\leq \|\hat{\Sigma}_S^{-1} - \Sigma^{-1}\|\cdot\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\| + \|\Sigma^{-1}\|\cdot\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\| \\
&\leq \left(\frac{\|\Sigma^{-1}\|^2\cdot\|\hat{\Sigma}_S - \Sigma\|}{1-\|\hat{\Sigma}_S^{-1}\|\cdot\|\hat{\Sigma}_S-\Sigma\|} + \|\Sigma^{-1}\|\right)\cdot\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\| \\
&= \frac{\|\Sigma^{-1}\|}{1-\|\hat{\Sigma}_S^{-1}\|\cdot\|\hat{\Sigma}_S-\Sigma\|}\cdot\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\| \\
&\leq \nu^{-1}\|\Sigma^{-1}\|\cdot\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\|.
\end{aligned}$$

The second inequality comes from Lemma 2. Let us first consider $\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}$. Recall that $\boldsymbol{\varepsilon} \sim \mathrm{subG}_d(\sigma^2)$, $\sup_{\mathbf{x}_j\in\mathcal{X}}\|\mathbf{x}_j\| \leq 1$, then the random variable $\mathbf{x}_j^\top\boldsymbol{\varepsilon} \sim \mathrm{subG}_d(\sigma^2)$. Denote by $\epsilon = \max_{j,k}|\hat{\Sigma}_{j,k} - \Sigma_{j,k}|$, we have

$$\begin{aligned}
\mathbb{P}(\|\mathbf{X}_S^\top\boldsymbol{\varepsilon}/n_{sub}\| \geq \epsilon) &\leq \sum_{i=1}^d \mathbb{P}(|\mathbf{x}_i^\top\boldsymbol{\varepsilon}| \geq n_{sub}\epsilon/\sqrt{d}) \\
&\leq 2d\exp\left(-\frac{n_{sub}\epsilon^2}{2d\sigma^2}\right).
\end{aligned}$$

According to the fact that $\|\hat{\Sigma}_S - \Sigma\| \leq d\epsilon$. It follows from Lemma 3 that

$$\mathbb{P}(\|\hat{\Sigma}_S - \Sigma\| \geq \epsilon) \leq 2d^2\exp\left(-\frac{n_{sub}\epsilon^2}{2d^2\alpha(\lambda_r\vee 0)}\right).$$

Given that $\|\Sigma^{-1}\|\cdot\|\hat{\Sigma}_S - \Sigma\| \leq 1-\nu \in (0,1)$, by taking $\nu = 1/2$, we know

$$\begin{aligned}
\mathbb{P}(\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| \geq \epsilon) &\leq \mathbb{P}\left(\|\Sigma^{-1}\|\cdot\|\hat{\Sigma}_S - \Sigma\| \geq \frac{1}{2}\right) \\
&\quad + \mathbb{P}\left(\|\Sigma^{-1}\|\cdot\|\mathbf{X}_S^\top\varepsilon/n_{sub}\| \geq \frac{\varepsilon}{2}\right) \\
&\leq 2d^2 e^{-\frac{n_{sub}\alpha(\lambda_r\vee 0)^{-1}}{8d^2\|\Sigma^{-1}\|^2}} + 2de^{-\frac{n_{sub}\epsilon^2}{8d\sigma^2\|\Sigma^{-1}\|^2}}.
\end{aligned}$$

Then

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| \leq \sigma\|\Sigma^{-1}\|^2\sqrt{\frac{8d\log(d/\delta)}{n_{sub}}}$$

holds with confidence at least $1-\delta$ if $n_{sub} \geq 8\alpha(\lambda_r \vee 0)d^2\|\Sigma^{-1}\|^2\log(d^2/\delta)$. This completes the proof. ■

Remark 3 *Theorem 1 indicates that the GMS estimator $\hat{\boldsymbol{\beta}}$ is consistent, i.e. $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| \to 0$ as $n_{sub} \to \infty$. The founding condition requires that the right spectral gap of the invariant distribution π satisfies $1-\lambda_r > 0$. GMS almost trivially satisfies this condition. To see this, notice that the Markov chain generated by GMS is uniformly ergodic, and hence geometrically ergodic. Following this, it is known that $\lambda(P) < 1$ if and only if the Markov chain is geometrically ergodic (Roberts and Rosenthal 1997), so the condition is satisfied. The convergence rate is with order of $\mathcal{O}(\sqrt{\frac{d\log(d)}{n_{sub}}})$, which matches the optimal error bound for the i.i.d. samples in a minimax sense.*

Denote the mean squared error of the prediction $\mathbf{X}\hat{\boldsymbol{\beta}}$ by $\mathrm{MSE}(\mathbf{X}\hat{\boldsymbol{\beta}}) = \frac{1}{n}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|^2$. According to the inequality

$$\mathrm{MSE}(\mathbf{X}\hat{\boldsymbol{\beta}}) \leq \lambda_{\max}(\Sigma_n)\cdot\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|^2,$$

we can obtain the following corollary bounding the prediction error immediately from Theorem 1.

Corollary 1 *Under the same conditions in Theorem 1, with confidence at least $1-\delta$, it holds*

$$\mathrm{MSE}(\mathbf{X}\hat{\boldsymbol{\beta}}) \leq C^2\lambda_{\max}(\Sigma_n)\cdot\frac{8d(\log(d)+\log(1/\delta))}{n_{sub}}. \quad (7)$$

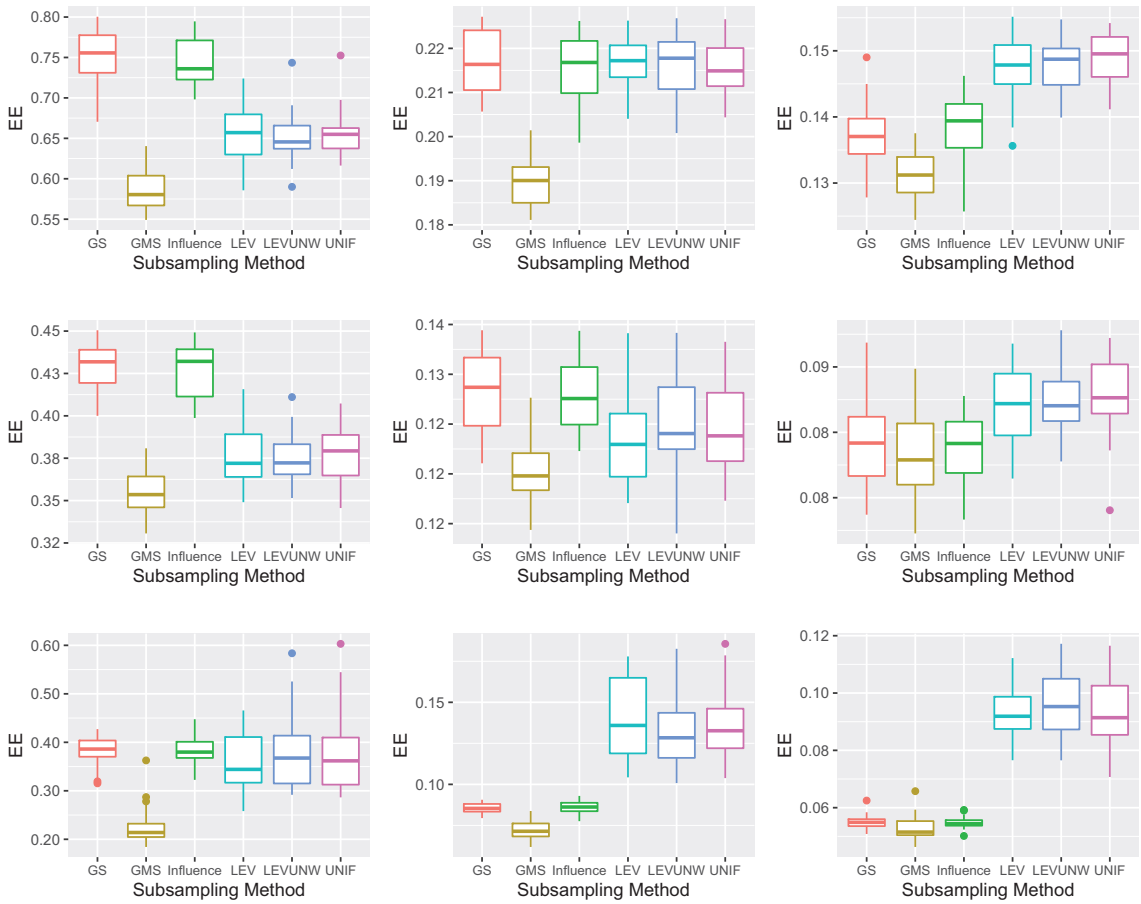


Figure 3: Boxplots of 50 times experiments for different subsampling methods with $n = 1M, d = 500$. From top to bottom: $M1(\mathbf{N}), M1(\mathbf{U}), M1(\mathbf{t})$. From left to right: $sr = 0.001, 0.005, 0.01$.

Experiments

To assess the performance of GMS, we conduct experiments on both simulation studies and real data examples. All numerical studies are conducted in software R on Compute Canada clusters with 2.1 GHz CPUs and 128 GB memory.

Simulation Studies

In simulation studies, we generate the data by $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$, where the $n \times d$ design matrix $\mathbf{X}$ is generated by a mixture of Gaussian distributions $\frac{1}{2}\mathbf{N}(\mu_1, \sigma_1^2) + \frac{1}{2}\mathbf{N}(\mu_2, \sigma_2^2)$ in two different ways: $(M1)$ $\mu_1 = -2, \sigma_1 = 3$, $\mu_2 = 2, \sigma_2 = 10$; $(M2)$ $\mu_1 = 0, \sigma_1 = 3$, $\mu_2 = 0, \sigma_2 = 10$. The oracle $\boldsymbol{\beta}^*$ is generated uniformly from $\{\pm 3, \pm 2, \pm 1, 0\}$. We generate three different types of i.i.d. noise, including uniform distribution with $\varepsilon_i \sim \mathbf{U}(-5, 5)$, normal distribution with $\varepsilon_i \sim \mathbf{N}(0, 25)$ and Student-t distribution with $\varepsilon_i \sim \mathbf{t}(2)$. We denote the models combining these design matrices and noise distributions as follows: $M1(\mathbf{U}), M1(\mathbf{N}), M1(\mathbf{t}), M2(\mathbf{U}), M2(\mathbf{N}), M2(\mathbf{t})$. Also, we set $n = 100K, 500K, 1M$ and with corresponding $d = 50, 250, 500$.

For each dataset, we compare the proposed GMS with five representative sampling methods where each method is applied $K = 50$ times repeatedly. The quality of the fit is measured by the estimation error (EE):

$$\text{EE} = \frac{1}{K}\sum_{k=1}^{K} \|\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}^*\|.$$

In all experiments, the subsample size is set by $n_{sub} = sr * n$, where sr represents the sampling ratio. We set $sr = 0.001, 0.005, 0.01$ for each model. If required, a pilot estimator is calculated by uniform subsampling of size $n_0 = n_{sub}$. The sampling methods considered for comparison are: GS, leverage subsampling (LEV), unweighted leverage subsampling (LEVUNW) (Ma, Mahoney, and Yu 2015), uniform sampling and influence-based sampling (Ting and Brochu 2018). Note that LEVUNW conducts sampling identically to LEV, but solves the unweighted least squares problem instead. For influence-based sampling, the sampling weight for $(\mathbf{x}_i, \mathbf{y}_i)$ is proportional to $\|\psi_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i)\|$, where

$$\psi_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) = (y_i - \mathbf{x}_i\boldsymbol{\beta})\Sigma_n^{-1}\mathbf{x}_i$$

is the influence function.

Table 1: EE comparison (mean $\pm$ standard deviation)$\times 10^{-2}$ for different sampling methods for $n = 1M, d = 500$

Methods	$M1(\mathbf{N})$			$M1(\mathbf{U})$			$M1(\mathbf{t})$		
	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$
GS	74.9(3.89)	21.7(0.68)	13.7(0.45)	42.9(1.36)	12.8(0.40)	8.33(0.26)	38.2(2.99)	8.54(0.31)	5.48(0.24)
Influence	74.5(2.97)	21.5(0.78)	13.8(0.50)	42.8(1.54)	12.8(0.35)	8.30(0.21)	38.4(3.13)	8.61(0.36)	5.47(0.19)
LEV	65.4(3.41)	21.7(0.52)	14.7(0.50)	37.6(1.64)	12.3(0.42)	8.57(0.22)	35.4(5.71)	14.1(2.51)	9.26(0.98)
LEVUWN	65.0(2.81)	21.7(0.73)	14.7(0.40)	37.4(1.47)	12.5(0.46)	8.59(0.20)	37.8(7.52)	13.2(2.25)	9.72(1.25)
UNIF	65.5(2.72)	21.6(0.66)	14.9(0.37)	37.8(1.52)	12.5(0.45)	8.64(0.25)	37.8(8.23)	13.6(2.15)	9.36(1.26)
GMS	**58.6**(2.36)	**18.9**(0.61)	**13.1**(0.36)	**35.5(1.35)**	**12.0**(0.31)	**8.26**(0.23)	**22.5**(3.65)	**7.22**(0.50)	**5.30**(0.22)

Table 2: EE comparison (mean $\pm$ standard deviation)$\times 10^{-2}$ for different sampling methods for $n = 1M, d = 500$

Methods	$M2(\mathbf{N})$			$M2(\mathbf{U})$			$M2(\mathbf{t})$		
	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$
GS	77.8(2.55)	22.2(0.67)	14.4(0.43)	43.3(1.65)	13.4(0.41)	8.62(0.24)	39.3(3.71)	8.95(0.48)	5.66(0.34)
Influence	77.4(2.53)	22.3(0.49)	14.4(0.39)	44.0(1.39)	13.4(0.36)	8.73(0.33)	39.5(3.38)	9.00(0.51)	5.68(0.36)
LEV	67.3(3.11)	22.4(0.61)	15.5(0.42)	39.3(1.26)	13.0(0.45)	8.89(0.21)	36.1(5.19)	14.1(2.22)	9.52(1.05)
LEVUWN	67.6(2.85)	22.3(0.67)	15.4(0.37)	38.7(1.52)	12.9(0.43)	8.93(0.22)	37.0(6.46)	13.2(1.39)	13.0(2.73)
UNIF	68.5(3.36)	22.5(0.64)	15.4(0.47)	38.8(1.52)	13.0(0.42)	8.81(0.31)	35.8(4.48)	13.5(2.03)	9.60(1.42)
GMS	**60.5**(2.68)	**19.4**(0.65)	**13.4**(0.46)	**37.0**(1.53)	**12.6**(0.43)	**8.61**(0.26)	**22.7**(1.78)	**7.52**(0.66)	**5.65**(0.88)

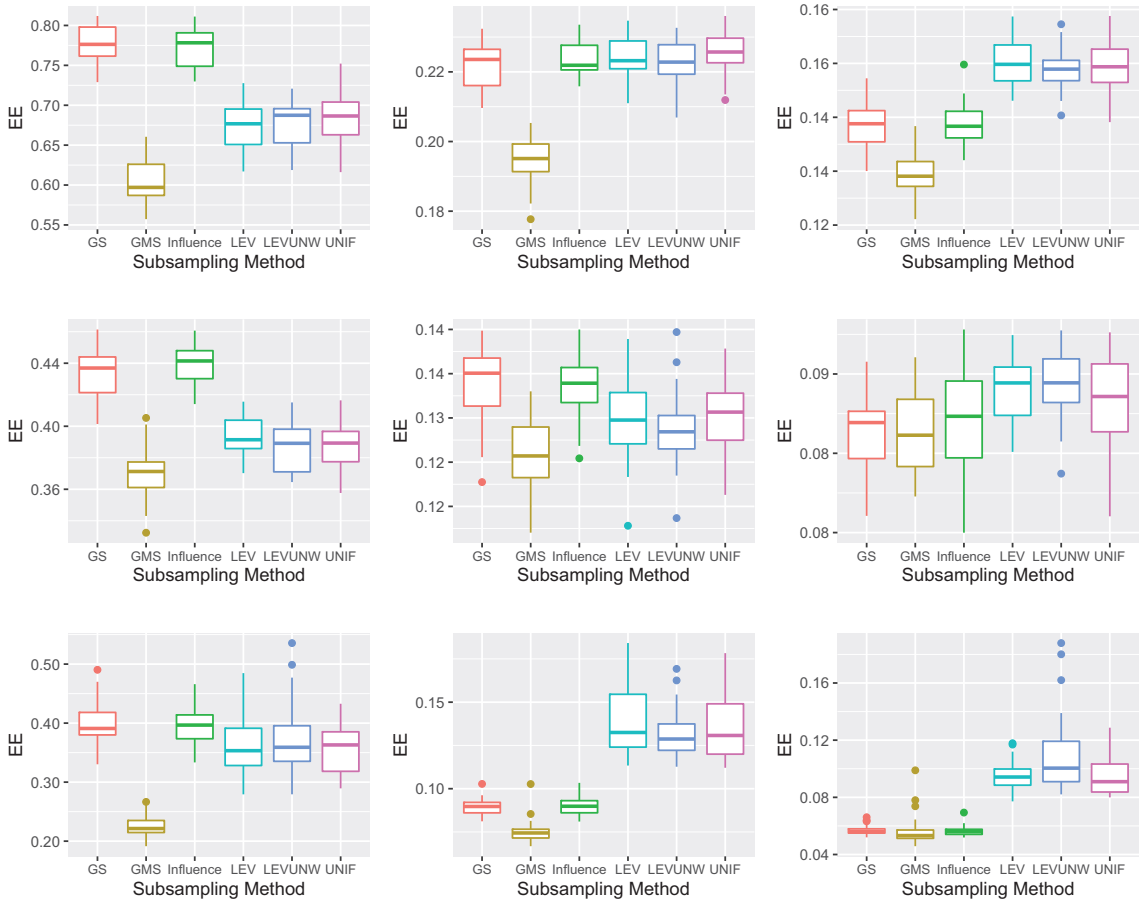


Figure 4: Boxplots of 50 times experiments for different subsampling methods with $n = 1M, d = 500$. From top to bottom: $M2(\mathbf{N}), M2(\mathbf{U}), M2(\mathbf{t})$. From left to right: $sr = 0.001, 0.005, 0.01$.

Due to space limitation, we only show the results for the setting $n = 1M, d = 500$. Other results are given in the

Table 3: EE comparison (mean $\pm$ standard deviation) for different sampling methods for real datasets

Methods	Online News Popularity			Poker Hands			Wave Energy Converters		
	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$	$sr = 0.1\%$	$sr = 0.5\%$	$sr = 1\%$
GS	36.4(26.7)	8.67(5.27)	7.06(4.08)	0.44(0.18)	**0.118**(0.039)	**0.078**(0.030)	1979(258.6)	767(105.1)	**498**(74.8)
Influence	29.8(31.9)	8.02(5.15)	**5.22**(2.06)	0.41(0.17)	0.125(0.036)	0.079(0.023)	1990(339.1)	**696**(96.1)	503(51.9)
LEV	18.6(19.1)	11.6(7.01)	8.55(4.55)	0.35(0.13)	0.12(0.04)	0.092(0.024)	1940(275.5)	801(99.1)	579(66.2)
LEVUWN	21.7(13.1)	10.3(8.38)	8.85(4.92)	0.37(0.13)	0.123(0.03)	0.078(0.023)	1868(239.8)	808(86.8)	583(66.1)
UNIF	18.1(14.2)	12.4(7.75)	7.23(3.01)	0.35(0.12)	0.135(0.035)	0.089(0.022)	1956(278.1)	815(90.5)	563(76.8)
GMS	**11.4**(5.24)	**7.48**(4.81)	5.37(2.61)	**0.26**(0.09)	0.119(0.034)	0.079(0.024)	**1747**(269.1)	762(103.9)	517(67.1)

supplementary material. Figs. 3 and 4 record the boxplots based on 50 times empirical estimation error. The mean and standard deviation of EE are reported in Tables 1 and 2. Several observations are worth making about the presented results. To begin, as the subsample size increases, the mean error and standard deviation for all methods tend to decrease monotonically. Note that GMS achieves the lowest error for all 6 models under all sampling ratios. In particular, GMS outperforms other competitors significantly when the sampling ratio is very small. This observation supports that the samples generated by GMS are more informative and lead to robust estimation. Moreover, LEV and LEVUWN perform quite similarly to uniform sampling. This is because the leverage score only depends the input information. Since both $M1$ and $M2$ are generated by mixtures of Gaussian and hence have nearly uniform leverage scores, LEV and LEVUWN do not show significant differences with uniform sampling. We also observe that influence-based subsampling performs almost identically to GS. Since the design matrix $\mathbf{X}$ is generated by i.i.d. mixtures of Gaussian, Σ_n approximates a diagonal matrix. Therefore, the influence function assigns similar sampling probability as gradient in the sampling process. In addition, in heavy-tailed noise cases, i.e. $M1(t), M2(t)$, GMS still achieves the lowest error, which supports that GMS is more robust to highly noisy data.

Real Data Examples

We further evaluate the performance of GMS on 3 real-world datasets: Online News Popularity ($n = 39797, d = 61$), Wave Energy Converters ($n = 288000, d = 32$) and Poker Hands ($n = 25010, d = 11$) [1]. For the WEC dataset, we remove 16 columns due to collinearity. Since the oracle $\boldsymbol{\beta}^*$ is unknown for real datasets, we utilize $\boldsymbol{\beta}_n$ as a proxy to $\boldsymbol{\beta}^*$ in our performance metric. The comparisons of empirical estimation error with different sampling ratio are reported in Table 3. It can be observed from Table 3 that GMS still achieves the lowest error when the sampling ratio is very low. As subsample size increases, GMS is still able to achieve competitive performance among the 6 sampling algorithms.

Conclusion

In this paper, we propose a gradient-based Markov subsampling (GMS) algorithm for the linear regression problem. We analyze the performance of GMS in terms of error bounds. The theoretical results show that the GMS estimator is statistically consistent and the corresponding error bound matches the optimal rate in the minimax sense. Experiments on simulation studies and real data examples demonstrate the effectiveness of GMS. Future work includes extending GMS to general models (e.g. Ridge, Lasso), and accelerating the burn-in process of GMS. All these problems are under current research.

[1] https://archive.ics.uci.edu/ml/datasets.php

Acknowledgments

This work was supported in part by NSERC grant RGPIN-2016-05024 and in part by NSFC grants 11690014 and 11971173. The content is solely the responsibility of the authors and does not necessarily represent the official views of the aforementioned funding agencies.

References

Burke, J. V.; Curtis, F. E.; Lewis, A. S.; Overton, M. L.; and Simões, L. E. 2018. Gradient sampling methods for nonsmooth optimization. *arXiv preprint arXiv:1804.11003*.

Burke, J. V.; Lewis, A. S.; and Overton, M. L. 2005. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization* 15(3):751–779.

Dereziński, M.; Warmuth, M. K.; and Hsu, D. 2018a. Correcting the bias in least squares regression with volume-rescaled sampling. *arXiv preprint arXiv:1810.02453*.

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

Down, D.; Meyn, S. P.; and Tweedie, R. L. 1995. Exponential and uniform ergodicity of markov processes. *The Annals of Probability* 23(4):1671–1691.

Drineas, P.; Mahoney, M. W.; Muthukrishnan, S.; and Sarlós, T. 2011. Faster least squares approximation. *Numerische mathematik* 117(2):219–249.

Drineas, P.; Magdon-Ismail, M.; Mahoney, M. W.; and Woodruff, D. P. 2012. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research* 13(Dec):3475–3506.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2008. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications* 30(2):844–881.

Fan, J.; Jiang, B.; and Sun, Q. 2018. Hoeffding's lemma for markov chains and its applications to statistical learning. *arXiv preprint arXiv:1802.00211*.

Fithian, W., and Hastie, T. 2014. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics* 42(5):1693–1724.

Gong, T.; Zou, B.; and Xu, Z. 2015. Learning with ℓ_1-regularizer based on markov resampling. *IEEE Transactions on Cybernetics* 46(5):1189–1201.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* 53(2):217–288.

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

Moritz, P., and et al. 2018. Ray: A distributed framework for emerging ai applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation (OSDI, 18)*, 561–577.

Roberts, G., and Rosenthal, J. 1997. Geometric ergodicity and hybrid markov chains. *Electronic Communications in Probability* 2:13–25.

Rudi, A.; Calandriello, D.; Carratino, L.; and Rosasco, L. 2018. On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, 5672–5682.

Rudolf, D. 2011. Explicit error bounds for markov chain monte carlo. *arXiv preprint arXiv:1108.3201*.

Sun, Q.; Tan, K. M.; Liu, H.; and Zhang, T. 2017. Graphical nonconvex optimization for optimal estimation in gaussian graphical models. *arXiv preprint arXiv:1706.01158*.

Sun, T.; Sun, Y.; and Yin, W. 2018. Explicit error bounds for markov chain monte carlo. *arXiv preprint arXiv:1809.04216v1*.

Ting, D., and Brochu, E. 2018. Optimal subsampling with influence functions. In *Advances in neural information processing systems*, 3650–3659.

Vershynin, R. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press.

Zaharia, M.; Chowdhury, M.; Franklin, M. J.; Shenker, S.; and Stoica, I. 2010. Spark: Cluster computing with working sets. *HotCloud* 10(10-10):95.

Zhu, R. 2016. Gradient-based sampling: an adaptive importance sampling for least-squares. In *Advances in neural information processing systems*, 406–414.