MASSIVE DATA K-MEANS CLUSTERING AND BOOTSTRAPPING VIA A-OPTIMAL SUBSAMPLING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Dali Zhou

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Fei Tan, Co-chair

Department of Mathematical Sciences

Dr. Hanxiang Peng, Co-chair

Department of Mathematical Sciences

Dr. Benzion Boukai

Department of Mathematical Sciences

Dr. Jyotirmoy Sarkar

Department of Mathematical Sciences

Dr. Peijun Li

Department of Mathematics, Purdue University

Approved by:

Dr. Eugene Mukhin

Head of the Graduate Program,

Department of Mathematical Sciences, IUPUI

This thesis is dedicated to my parents.

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisors Professor Hanxiang Peng and Professor Fei Tan with great appreciation from the bottom of my heart. During these many years of my Ph.D life, you taught me not only knowledge, professional skills and how to do research, but also the correct attitude to be a good man. I cannot thank you enough for what you have done during the most influential 6 years in my life. As an international student, I feel you are my family in the U.S..

My sincere thank you also goes to Professor Benzion Boukai and Professor Jyoti Sarkar. I learned so much from you in class and out of class. I still remember how I learned to construct hypothesis tests in STAT528, and how I started coding in R and presenting projects in STAT533. You are not only my teachers but also my friends. When you agreed to be my dissertation committee members, I felt greatly honored. I also would like to express my great thanks to Professor Peijun Li from Purdue University, who generously devotes his time to be a member of my dissertation committee.

Besides, I would like to thank Professor Fang Li and Professor Honglang Wang who taught me and instructed me in classes and graduate life.

The most thank you goes to my parents Yuewen Zhou, Kuijing Jiang and my wife Ye Li, who have always supported my Ph.D dream. Without your support, I can not be what I am today. I love you!

Last but not least, I would like to thank the department of mathematical sciences where I study, work and get support. I want to thank our research team and the friends I made at IUPUI who helped me in balancing my Ph.D study life. Besides, I would like to thank Heng Xu and Hopi Lin, who guided me in my career path.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABBREVIATIONS

CDF	Cumulative Density Function
i.i.d.	independent and identically distributed
LLN	Law of Large Number
LSE	Least Square Estimate
MSE	Mean Squared Error
NLP	Natural Language Processing
OLSE	Ordinary Least Square Estimate
o.w.	otherwise
PDF	Probability Density Function
r.v.	random variable
SLLN	Strong Law of Large Number
WLSE	Weighted Least Square Estimate
w.r.t	with respect to
WSS	Within Sum of Squares

ABSTRACT

Zhou, Dali Ph.D., Purdue University, August 2019. Massive Data K-means Clustering and Bootstrapping via A-optimal Subsampling. Major Professors: Hanxiang Peng and Fei Tan.

For massive data analysis, the computational bottlenecks exist in two ways. Firstly, the data could be too large that it is not easy to store and read. Secondly, the computation time could be too long. To tackle these problems, parallel computing algorithms like Divide-and-Conquer were proposed, while one of its drawbacks is that some correlations may be lost when the data is divided into chunks. Subsampling is another way to simultaneously solve the problems of the massive data analysis while taking correlation into consideration. The uniform sampling is simple and fast, but it is inefficient, see detailed discussions in Mahoney (2011) and Peng and Tan (2018). The bootstrap approach uses uniform sampling and is computing time intensive, which will be enormously challenged when data size is massive. *k*-means clustering is standard method in data analysis. This method does iterations to find centroids, which would encounter difficulty when data size is massive. In this thesis, we propose the approach of optimal subsampling for massive data bootstrapping and massive data *k*-means clustering. We seek the sampling distribution which minimize the trace of the variance co-variance matrix of the resulting subsampling estimators. This is referred to as A-optimal in the literature. We define the optimal sampling distribution by minimizing the sum of the component variances of the subsampling estimators. We show the subsampling *k*-means centroids consistently approximates the full data centroids, and prove the asymptotic normality using the empirical process theory. We perform extensive simulation to evaluate the numerical performance of the proposed optimal subsampling approach through the empirical MSE and the running times. We also applied the subsampling approach to real data. For massive data bootstrap, we conducted a large simulation study in the framework of the linear regression based on the A-optimal theory proposed by Peng and Tan (2018). We focus on the performance of confidence intervals computed from A-optimal subsampling, including coverage probabilities, interval lengths and running times. In both bootstrap and clustering we compared the A-optimal subsampling with uniform subsampling.

1. INTRODUCTION

1.1 K-means Clustering

Interests in partitioning of objects has risen in different fields, like statistics, computer science and their intersect area: machine learning. There are two types of models in machine learning: supervised learning models (in statistics we call classification models) and unsupervised learning models (in statistics we call clustering models). k-means is a popular model of unsupervised learning. In the year of 1954, an anthropological data analysis article (JSTOR) firstly used "cluster analysis", which at that time was called "grouping".

Developed from signal processing originally, k-means clustering partitions the data set into desired number of clusters, where each observation is assigned to the cluster with nearest centroid (mean in general). The idea came from Steinhaus, the early father of data science, in 1956. It was then firstly named "k-means" and fully introduced by MacQueen (1967), some consistency results were also provided. Hartingan (1978) proved a central limit theorem and convergence in probability for partitioning one dimensional data into two clusters. Pollard (1981,1982) extended the results and gave strong consistency results and a central limit theorem for multidimensional case.

Even today, after 50 years of development of k-means clustering, there are still much left for us to dig in. Steinley (2006) reviewed the problems solved and unsolved in k-means, some challenges like how to choose initial centroids and how to determine the number of clusters k were pointed out. Besides theoretical research from statisticians, to develop the most efficient algorithm of k-means has been also an important topic in computer science. Bock (2008) discussed the original algorithms and extensions for k-means clustering analysis. Jain (2009) concluded the k-means data clustering and beyond, also pointed out the challenge of large-scale clustering in this era of big data.

To implement k-means, several algorithms have been developed. The standard algorithm is the EM algorithm, developed by Lloyd (1957, published 1982). Other popular algorithms are the MacQueen (1967) algorithm, the Hartigan & Wong (1979) algorithm and Elkan (2003) triangle inequality algorithm.

1.2 Bootstrapping

Bootstrap, introduced by Bradley Efron (1979) is a resampling method for obtaining statistical properties of estimates. It is a widely used method in different area of statistics, traditionally in which the number of observations is too small that large sample theories can not be applied (for example in clinical trial studies when sample size is normally small because of expense), or in cases where the explicit theoretical results are too complicated to be obtained. In last century, Bootstrap has become a popular tool used to obtain variance, bias, confidence region of the estimators. As artificial intelligence and data science become more and more important in today's world, bootstrap is also becoming important in machine learning and other computing areas, for example, in cross validation to prevent over fitting problems. Or more generally, when the resampling sample size is not the same as original sample size (which is called m-out-of-n bootstrap, Bickel (1997)), bootstrap can be used in big data analysis when taking m much smaller than n.

The spirit of bootstrap method is to take a resample from the original data to calculate the sampling distribution of the estimator that we are interested in. The relationship between the resample and original sample can be used to mimic the relationship between the sample and population. That is, by treating sample as the "population", resample as the "sample", we can take "sample" from the "population" repeatedly to obtain a sampling distribution of the resampling estimator. Because of the consistency of resampling estimator, we will be able to calculate standard error of the estimator and construct confidence region of of the true parameter. This is supported by the asymptotic theories provided by Bickel and Freedman (1981), and Singh (1981). Better bootstrap confidence intervals with an improvement that results in second order correctness were provided by Efron (1987). The basic bootstrap theoretical results, and their applications were mainly included in the books written by Efron and Tibshirani (1994), Shao and Tu (1995) and Davison and Hinkley (1997).

There are different variations of bootstrap. Since regression models are essential models in the statistics world. Bootstrap method is also applied to this simple while powerful model in approximating the sampling distribution of regression estimators. Freedman (1981) showed the validness of bootstrap approximation to the distribution of regression least squares estimators. Wu (1986) studied resampling methods including jackknife and bootstrap in regression models. There are basically two types of bootstrap in regression models: paired bootstrap and residual bootstrap. Sometimes they are also called resampling bootstrap and model-based bootstrap in regression models, respectively.

Bootstrap is uniform resampling, which treats all the observations equally likely. It could be therefore generalized to non-uniform resampling, which people also call weighted bootstrap or generalized bootstrap. Bayesian bootstrap introduced by Rubin (1981) is one type of weighted bootstrap, in which each observation is assigned random resampling weight. Charterjee and Bose (2005) discussed theoretical results of generalized bootstrap for estimating equations. In their paper, assumptions of the weights and some examples of weights are given. For example, jackknife, delete-d jackknife, m out of n bootstrap can all be considered as special cases of weighted bootstrap. However, these weights above do not improve the bootstrap in the sense of efficiency.

Typically, bootstrap is used for data with small sample size. As the fast development of computer science, larger and larger data sets are generated and stored. How to deal with large scaled data has now become a crucial problem. Bootstrap methods are also generalized to big data applications. Kleiner, el (2012) proposed a scalable bootstrap for massive data, Bag of Little Bootstraps (BLB) to improve robustness.

1.3 A-optimal Subsampling in Big Data Analysis

Resampling techniques were developed last century but somewhat limited by the computing ability of computers. With the fast development of technology, the computationally intensive statistical resampling methods are brought back to the stage even for large data sets.

In big data analysis, subsampling method is popular in solving the problems that are hard for traditional models or computers. Normally, researchers use uniform subsampling as a way to increase computing efficiency. However, this way of subsampling treats all the observations equally likely, while different observations could be of different levels of importance. To settle this issue, statisticians choose to do weighted subsampling, finding the weights of observations before statistical analysis, in this way, the weighted subsample will contain more information than uniform subsample. Hence, a more efficient and accurate estimator could be obtained.

Drineas *et al*(2006) introduced a weight for approximating matrix multiplication. Ping Ma, *et al* (2015) introduced a sampling weight for regression models using leverage score, which works better than uniform subsample. Rong Zhu, *et al*(2015) developed optimal subsampling approaches for large sample linear regression, by minimizing the trace of center of certain matrix. Peng and Tan (2018) improved their result and developed the A-optimal sampling weights. Below are two examples of A-optimal subsampling use.

1.3.1 Example: Matrix Multiplication Approximation via optimal Subsampling

Drineas (2006) proposed the optimal sampling method for approximating matrix multiplication. Here we perform an example to illustrate how optimal subsampling is betting than uniform subsampling. To be specific, given two input matrices $\mathbf{A}_{l\times n}$ and $\mathbf{B}_{n\times p}$, to approximate the product $\mathbf{AB}$ in an efficient way, firstly, form matrices $\mathbf{C}_{l\times c}$ and $\mathbf{R}_{c\times p}$ by sampling c columns of $\mathbf{A}$ with appropriate probability $\mathbf{P}$ on $\{1, 2, ..., n\}$ and using the same c rows of $\mathbf{B}$. Then scale the sampled columns and rows appropriately.

Algorithm 1: Optimal Subsampling Algorithm for Matrix Multiplication Approximation

```
Input  : $\mathbf{A} \in \mathbb{R}^{l\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ and $1 \le c \le n$.
Output: $\mathbf{C} \in \mathbb{R}^{l\times c}$ and $\mathbf{R} \in \mathbb{R}^{c\times p}$.
for $t \in \{1, \ldots, c\}$ do
    Calculate sampling probabilities $p_k$, $k = 1, 2, ..., n$.
    Pick $i_t \in \{1, ..., n\}$ using probabilities $p_1, ..., p_n$
      independently with replacement;
    Let $\mathbf{C}^{(t)} = \mathbf{A}^{(i_t)}/\sqrt{cp_{i_t}}$ and $\mathbf{R}_{(t)} = \mathbf{B}_{(i_t)}/\sqrt{cp_{i_t}}$,
      where $\mathbf{C}^{(t)}$ denote the $t$th column of $\mathbf{C}$, and $\mathbf{R}_{(t)}$
      denote the $t$th row of $\mathbf{R}$;
end
Output $\mathbf{CR}$.
end
```

The output product $\mathbf{CR}$ is an approximation of the matrix product $\mathbf{AB}$. In this algorithm, the choice of p_k and the scaling of column and row are important features that could make it a good approximation. In fact,

$$p_k = \frac{|\mathbf{A}^{(k)}||\mathbf{B}_{(k)}|}{\sum_{k'=1}^{n} |\mathbf{A}^{(k')}||\mathbf{B}_{(k')}|}$$

was proved to be the optimal choice that minimizes $\mathbf{E}\Big[||\mathbf{AB} - \mathbf{CR}||_F^2\Big]$, where the F-norm is define by

$$||\mathbf{AB} - \mathbf{CR}||_F = \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{p}(\mathbf{AB} - \mathbf{CR})_{ij}^2}$$

As can be seen, the spirit here is to minimize the norm of certain matrix to get the optimal probability, which just provide us an idea on how to choose the appropriate weight in the optimal subsampling.

To compare the optimal sampling method to uniform sampling in matrix multiplication approximation, a simulation example is constructed as below.
Let $\mathbf{A}$ be a $18 \times n$ matrix, $\mathbf{B}$ be a $n \times 60$ matrix. The values of n are chosen to be 8, 16, 40, 100. For each given n, the values of r are selected to be $0.3 * n,\ 0.5 * n,\ 0.8 * n$ and n (all are rounded down to the nearest integer). Three different types of matrices are generated:

- **Uniform Matrix**, all elements of $\mathbf{A}$ and $\mathbf{B}$ are generated from $Unif(0, 1)$, a uniform distribution with parameters 0 and 1.

- **Mixture Matrix**, elements of $\mathbf{A}$ are evenly generated from 6 different distributions in the order of: $Unif(20, 21)$, $\mathcal{N}(-10, 1)$, $\mathcal{E}xp(10)$, $Unif(0, 1)$, $\mathcal{N}(-1000, 2)$ and $\mathcal{E}xp(1)$. Matrix $\mathbf{B}$ is generated from $N(-100, 1)$, $Unif(2000000, 2000001)$, $\mathcal{E}xp(100)$, $Unif(0, 1)$, $\mathcal{N}(1000, 2)$ and integer sequence from 1 to $10n$.

- **Heavy Mixture Matrix**, generated similarly with **Mixture Matrix** but with even more different distribution parameters.

The norm ratios of the following form are compared in table (1.1):

$$\frac{\mathbf{E}\big[||\mathbf{AB} - \mathbf{C}_{opt}\mathbf{R}_{opt}||_F^2\big]}{\mathbf{E}\big[||\mathbf{AB} - \mathbf{C}_{unif}\mathbf{R}_{unif}||_F^2\big]}.$$

From the output table, we can see that

- All the norm ratios are less than 1, which means the optimal subsampling is more statistically accurate in matrix multiplication approximation.

- In Uniform Matrix column, norm ratios are close to 1. The difference between two methods are not obvious. The reason is that the uniform matrices are incoherent, and uniform subsampling works well in this situation.

- In Mixture Matrix column, since the matrices have large coherence, the norm ratios are significantly less than 1 when n is small. As n becomes larger, the differences fade away but are still larger than those under uniform matrices situation.

- In Heavy Mixture Matrix column, the trend is more evident.

The simulation result shows that optimal subsampling method outperforms regular uniform subsampling in all cases and if the matrices have large coherence, or, if the matrices have quite different row norms, optimal subsampling method performs even better.

Table 1.1.: Comparison of Norm Ratios in Matrix Multiplication Approximation

n	**r**	**Uniform Matrix**	**Mixture Matrix**	**Heavy Mixture Matrix**
8	2	0.9994	0.7572	0.5102
	4	0.9999	0.7541	0.5660
	6	0.9999	0.7561	0.5669
	8	0.9999	0.7605	0.5612
16	4	0.9999	0.8938	0.7751
	7	1.0000	0.8931	0.7681
	12	1.0000	0.8933	0.7512
	15	1.0000	0.8937	0.7499
40	12	1.0000	0.9839	0.9177
	20	1.0000	0.9838	0.9223
	32	1.0000	0.9840	0.9200
	40	1.0000	0.9841	0.9217
100	30	1.0000	0.9970	0.9756
	50	1.0000	0.9970	0.9765
	80	1.0000	0.9970	0.9774
	100	1.0000	0.9970	0.9770

1.3.2 Example: Empirical Distribution Function Approximation via Optimal Subsampling

Empirical distribution function $F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[x_i \leq x]$, constructed using data from sample $\{x_1, x_2, ..., x_n\}$ is an estimate of the CDF. It is a widely used non-parametric function. In this example, we focus on finding the optimal sampling weight for approximating the empirical distribution function by minimizing certain term. Some interesting results are given in theorem 1.3.2.

Theorem 1.3.1 *Let* $F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[X_i \leq x]$ *be the empirical distribution function. Let* $F_r^*(x) = \frac{1}{r}\sum_{j=1}^{r} \frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^*}$ *be the subsampling empirical distribution based on a subsample* $\{X_1^*, X_2^*, ..., X_r^*\}$ *drawn according to the sampling distribution* $\{\pi_1, \pi_2, ..., \pi_n\}$ *on the data points. Then the optimal subsampling probabilities are*

$$\pi_i^o = \frac{\mathbf{1}[X_i \leq x]}{\sum_{i=1}^{n} \mathbf{1}[X_i \leq x]}, \quad i = 1, 2, ..., n.$$

Proof We will find the optimal sampling probabilities by minimizing the variance of F_r^* given the data $X_1, X_2, ..., X_n$. For convenience, we write $F_r^*(x)$ as F_r^* and write $F_n(x)$ as F_n in the proof. Also, for subsampling statistic T^*, use $E^*(T^*)$ and $V^*(T^*)$ to denote the conditional expectation $E(T^*|X_1, X_2, ..., X_n)$ and conditional variance $V(T^*|X_1, X_2, ..., X_n)$. We start with calculating the conditional expectation first.

$$\begin{aligned} E^*(F_r^*) &= E^*(\frac{1}{r}\sum_{j=1}^{r} \frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^*}) \\ &= \frac{1}{r}\sum_{j=1}^{r} E^*(\frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^*}) \\ &= E^*(\frac{\mathbf{1}[X_1^* \leq x]}{n\pi_1^*}) \\ &= \sum_{i=1}^{n} \frac{\mathbf{1}[X_i \leq x]}{n\pi_i} \cdot \pi_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[X_i \leq x] = F_n. \end{aligned}$$

Then the conditional variance given data $X_1, X_2, ..., X_n$ is

$$\begin{aligned} V^*(F_r^*) &= V^*(\frac{1}{r}\sum_{j=1}^{r}\frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^*}) \\ &= \frac{1}{r^2}V^*(\sum_{j=1}^{r}\frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^*}) \\ &= \frac{1}{r^2}\cdot rV^*(\frac{\mathbf{1}[X_1^* \leq x]}{n\pi_1^*}) \\ &= \frac{1}{r}[E^*[(\frac{\mathbf{1}[X_1^* \leq x]}{n\pi_1^*})^2] - E^{*2}(\frac{\mathbf{1}[X_1^* \leq x]}{n\pi_1^*})] \\ &= \frac{1}{r}[\sum_{i=1}^{n}\frac{\mathbf{1}[X_i \leq x]}{n\pi_i^2}\pi_i - F_n^2] \\ &= \frac{1}{r}[\sum_{i=1}^{n}\frac{\mathbf{1}[X_i \leq x]}{n\pi_i} - F_n^2]. \end{aligned}$$

With the restrictions $\sum_{i=1}^{n}\pi_i = 1$ and $\pi_i \geq 0$, we invoke the Lagrange multiplier and solve for the optimal π_i's. The Lagrange function is

$$L(\pi_1, ..., \pi_n, \lambda) = \frac{1}{r}\{\sum_{i=1}^{n}\frac{\mathbf{1}[X_i \leq x]}{n\pi_i} - F_n^2\} + \lambda(\pi_1 + ... + \pi_n - 1)$$

Take partial derivative w.r.t π_i,

$$\frac{\partial L(\pi_1, ..., \pi_n, \lambda)}{\partial \pi_i} = -\frac{1}{r}\cdot\frac{\mathbf{1}[X_i \leq x]}{n\pi_i^2} + \lambda = 0.$$

Solve the above equation with restrictions for π_i, we get the optimal subsampling probabilities:

$$\pi_j^o = \frac{\mathbf{1}[X_j \leq x]}{\sum_{i=1}^{n}\mathbf{1}[X_i \leq x]}, \quad j = 1, 2, ..., n.$$

■

Substituting the optimal subsampling probabilities obtained from Theorem 1.3.1, we can get the optimal empirical distribution function estimator

$$F_r^{o*}(x) = \frac{1}{r}\sum_{j=1}^{r}\frac{\mathbf{1}[X_j^* \leq x]}{n\pi_j^{o*}}.$$

We have the following result for $F_r^{o*}(x)$.

Theorem 1.3.2 *The empirical distribution function $F_n(x)$ is A-optimal, i.e., $F_r^{o*}(x) = F_n(x)$, $x \in \mathbb{R}$.*

Proof For subsample $X_1^*, X_2^*, ..., X_r^*$, the expression of the A-optimal subsampling probabilities that corresponds to the subsampling points are

$$\pi_j^* = \frac{\mathbf{1}[X_j^* \le x]}{\sum_{i=1}^n \mathbf{1}[X_i \le x]}, j = 1, 2, ..., r.$$

It is worthwhile to note that, there is no star on the denominator of π_j^* since the denominators of π_i, $i = 1, 2, ..., n$ are all the same and are constants given x and $X_1, X_2, ..., X_n$. In fact, we can write the optimal subsampling probabilities as

$$\pi_j^* = \frac{\mathbf{1}[X_j^* \le x]}{nF_n(x)}, j = 1, 2, ..., r.$$

Then,

$$\begin{aligned} F_r^*(x) &= \frac{1}{r}\sum_{j=1}^r \frac{\mathbf{1}[X_j^* \le x]}{n\frac{\mathbf{1}[X_j^* \le x]}{nF_n(x)}} \\ &= \frac{1}{r}\sum_{j=1}^r \frac{\mathbf{1}[X_j^* \le x]}{\mathbf{1}[X_j^* \le x]} \cdot F_n(x) \end{aligned}$$

Let $G_r^*(x) = \frac{1}{r}\sum_{j=1}^r \frac{\mathbf{1}[X_j^* \le x]}{\mathbf{1}[X_j^* \le x]}$. It is worth to note that the value of $\frac{\mathbf{1}[X_j^* \le x]}{\mathbf{1}[X_j^* \le x]}$ depends on x, it may not necessarily be 1 (could be defined as 0 when $x < \min(X_1^*, ..., X_r^*)$). Now we have

$$F_r^*(x) = G_r^*(x)F_n(x).$$

Let $X_{(i)}$ be the sorted sample points, $i = 1, 2, ..., n$, here we assume $F(x)$ is continuous so all the sample points are different. The discrete distribution case could be proved similarly when taking ties of sample points into consideration.

For $x < \min(X_1, X_2, ..., X_n)$, $F_n(x) = 0$, $\mathbf{1}[X_i \le x] = 0$, $i = 1, 2, ..., n_0$, no point will be drawn, thus $F_r^*(x) = 0 = F_n(x)$.

For $X_{(k-1)} \le x < X_{(k)}$, $k = 2, 3, ..., n$. We have $F_n(x) = \frac{k}{n}$, and

$$\pi_i = \frac{\mathbf{1}[X_i \le x]}{nF_n(x)} = \begin{cases} \frac{1}{k}, x \le X_{(k)} \\ 0, o.w. \end{cases}$$

Hence, only the sample points that are less than $X_{(k)}$ could be drawn, that being said, $\mathbf{1}[X_j^* < x] = 1$, $j = 1, 2, ..., r$. So $G_r^*(x) = \frac{1}{r}\sum_{j=1}^{r} 1 = 1$. $F_r^*(x) = G_r^*(x)F_n(x) = F_n(x)$

For $x \geq X_{(n)}$, $F_n(x) = 1$. Thus $\pi = \frac{1}{n}$, it becomes uniform sampling, and $\mathbf{1}[X^i < x] = 1$,$i = 1, 2, ..., n$, $\mathbf{1}[X_j^* < x] = 1$, $j = 1, 2, ..., r$. In this case, $G_r^*(x) = \frac{1}{r}\sum_{j=1}^{r} 1 = 1$.

So we have proved for all $x \in \mathbb{R}$, $F_r^*(x) = F_n(x)$.

■

1.4 Our Work

The rest of this thesis is organized as follows. In chapter 2, we discuss massive data k-means clustering via A-optimal subsampling. Consistency theorem and central limit theorem are given, the A-optimal sampling distribution is also given. In chapter 3, we discuss bootstrapping and propose massive data bootstrapping via A-optimal subsampling. In chapter 4, massive data simulation for both k-means and bootstrapping in A-optimal subsampling are performed and discussed. In chapter 5, we perform massive data k-means clustering via A-optimal subsampling in the applications of natural language processing.

2. K-MEANS CLUSTERING VIA A-OPTIMAL SUBSMAPLING

2.1 K-means Clustering

The k-means clustering method is a classic and popular clustering algorithm in machine learning. By minimizing the within cluster sum of squares, the centroids with minimized within cluster sum of squares are obtained from iterated algorithm. The distances of each observation to the centroids will then be calculated. The observations closest to a centroid will belong to the same cluster. The number of clusters is specified before the algorithm begins.

When sample size is large, the iteration in the clustering algorithm could be time consuming. In extreme cases, the minimizer of the within cluster sum of squares may not even be computable. To save computing time, or in the extreme case to make undoable problems doable, statisticians or data scientists will use subsampling method. A subsample with sample size substantially smaller than the original sample size can be obtained from uniform subsampling with replacement. In this case, researchers will be able to apply k-means algorithm much faster. However, the drawback of uniform subsampling is that it is not statistically efficient enough. Uniform subsampling method treats all the data points equally, instead of extracting information from observations with higher importance.

To improve the uniform subsampling method for massive data k-means algorithm, we propose the massive data k-means algorithm via A-optimal subsampling. In our method, we calculate the sampling probabilities for observations by minimizing certain matrix, then a subsample is obtained from the original data using the pre-calculated Probabilities. We will show that under this procedure we can get more stable centroids with smaller MSE. Two cases are studied: the equal cluster size case and unequal cluster size case. In both cases the MSEs of centroids under proposed method are smaller than those using uniform subsampling. Under unequal cluster size case, our method largely outperforms uniform subsampling.

2.1.1 K-means Clustering Algorithms

Independent observations $\mathbf{x}_1, ..., \mathbf{x}_n$ are made on the same probability distribution P on $\mathbb{R}^d$. In k-means procedure, the observations are partitioned into k clusters by minimizing the within cluster sum of squares. Equivalently, to get the best partition, we find a vector of centroids $\mathbf{b}_n = (\mathbf{b}_{n1}^\top, ..., \mathbf{b}_{nk}^\top)^\top \in \mathbb{R}^{kd}$ that minimizes the within cluster sum of squares

$$W_n(\mathbf{a}) = \frac{1}{n}\sum_{i=1}^{n} \min_{1\leq l\leq k} ||\mathbf{x}_i - \mathbf{a}_l||^2, \quad \mathbf{a} = (\mathbf{a}_1^\top, ..., \mathbf{a}_k^\top)^\top \in \mathbb{R}^{kd} \tag{2.1.1}$$

where k is the number of clusters, d is the dimension of each observation. To implement the method, MacQueen gave the algorithm of k-means in 1967, which is composed of steps below:

1. Select k points in the observation space as the initial cluster centroids.
2. For each observation, calculate the distances between that observation and the k centroids. Assign the observation to the cluster with the closest centroid.
3. For each cluster, Calculate the new centroid.
4. Repeat step 2 and step 3 until convergence criterion is met. A convergence criterion may be the norm of the difference of the centroids in the last two iteration being less than some prespecified small number.

To be more specific, the k-means algorithm is given below:

```
Algorithm 2: $k$-means Clustering Algorithm
  Input  : Data $\mathbf{X}_{n\times d} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\top$, $\mathbf{x}_i \in \mathbb{R}^d$. Number of
           clusters $k$.
  Output: Centroid vector $\mathbf{b}_n$, cluster label $\mathbf{y}$.
init
    Initialize cluster centroids $\mathbf{b}_{n1}^{(0)}, ..., \mathbf{b}_{nk}^{(0)}$ randomly;
repeat
    In iteration $t$, do the following steps:
    For each $\mathbf{x}_i$, set label $y_i^{(t)} := \arg\min_{1\leq l\leq k}||\mathbf{x}_i - \mathbf{b}_{nl}^{(t-1)}||^2$;
    For each $l$, set $\mathbf{b}_{nl}^{(t)} := \frac{\sum_{i=1}^{n}\mathbf{1}\{y_i^{(t)}=l\}*\mathbf{x}_i}{\sum_{i=1}^{n}\mathbf{1}\{y_i^{(t)}=l\}}$;
until Convergence criterion is met;
Output values from last iteration $t_l$, $\mathbf{b}_n = (\mathbf{b}_{n1}^{(t_l)\top}, ..., \mathbf{b}_{nk}^{(t_l)\top})^\top$
  and label $\mathbf{y} = (y_1^{(t_l)}, ..., y_n^{(t_l)})^\top$
end
```

Consider the case of massive sample size n, when performing k-means for full sample is too slow or even not doable, researchers perform the k-means clustering by uniform subsampling, which is to select a random subsample from the full sample with uniform sampling. In this way, every observation is treated equally likely. The algorithm is given below:

Algorithm 3: k-means Clustering Algorithm via Subsampling

```
Input  : Data $\mathbf{X}_{n\times d} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\top$, $\mathbf{x}_i \in \mathbb{R}^d$. Number of
         clusters $k$. Subsample size $r$;
Output: Centroid vector $\mathbf{b}^*$, cluster label $\mathbf{y}$;
subsampling
    Take a uniform subsample $\mathbf{X}^*_{r\times d} = (\mathbf{x}^*_1, ..., \mathbf{x}^*_r)^\top$ from $\mathbf{X}_{n\times d}$
     of size $r$ with replacement.
init
    Initialize cluster centroids $\mathbf{b}_1^{*(0)}, ..., \mathbf{b}_k^{*(0)}$ randomly;
repeat
    In iteration $t$, do the following steps:
    For each $\mathbf{x}^*_j$, $j = 1, 2, ..., r$, set label
      $y_j^{(t)} := \text{argmin}_{1\le l\le k} ||\mathbf{x}^*_j - \mathbf{b}_l^{*(t-1)}||^2$;
    For each $l$, set $\mathbf{b}_l^{*(t)} := \frac{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\}*\mathbf{x}^*_j}{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\}}$;
until Convergence criterion is met;
Output values from last iteration $t_l$: $\mathbf{b}^* = (\mathbf{b}_1^{*(t_l)\top}, ..., \mathbf{b}_k^{*(t_l)\top})^\top$
  and label $\mathbf{y} = (y_1^{(t_l)}, ..., y_n^{(t_l)})^\top$, where
  $y_i^{(t_l)} = \text{argmin}_{1\le l\le k} ||\mathbf{x}_i - \mathbf{b}_l^{*(t_l)}||^2$, $i = 1, 2, ..., n$
end
```

Since uniform subsampling procedure treats all the observations with equal importance, it does not extract important information from data, which may lead to inefficient result. Therefore, we perform the A-optimal subsampling. We will show that the estimator obtained from this way will have better properties than that from uniform subsampling.

2.2 K-means Clustering via A-optimal Subsampling

Let $\pi_1, ..., \pi_n$ denote a general weight distribution on the observations. Uniform weight $\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}$ is one special case of the general weight when all weights are equal.

Take a subsample $\mathbf{x}_1^*, ..., \mathbf{x}_r^*$ with size $r \leq\leq n$ using the weight distribution. We now approximate $\mathbf{b}_n$ by $\mathbf{b}^*$ which minimizes

$$\hat{W}_n(\mathbf{a}) = \frac{1}{n} \sum_{j=1}^{r} \frac{1}{r\pi_j^*} \min_{1 \leq l \leq k} ||\mathbf{x}_j^* - \mathbf{a}_l||^2, \mathbf{a} \in \mathbb{R}^{kd} \tag{2.2.1}$$

Our goal is to find the optimal weight such that the subsampling estimator (the optimal cluster centroid vector in our case) has higher accuracy and efficiency. A theorem about the optimal sampling distribution will be specified later, the explicit formula and properties will also be provided. Here we propose the algorithm to implement the method.

Algorithm 4: k-means Clustering Algorithm via A-optimal Subsampling

```
Input  : Data $\mathbf{X}_{n\times d} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\top$, $\mathbf{x}_i \in \mathbb{R}^d$. Number of
         clusters $k$. Subsample size $r$;
Output: Centroid vector $\mathbf{b}^*$, cluster label $\mathbf{y}$;
pre-calculation
    Calculate sampling distribution $\boldsymbol{\pi}$ for $\mathbf{X}_{n\times d}$;
subsampling
    Take a subsample $\mathbf{X}^*_{r\times d} = (\mathbf{x}^*_1, ..., \mathbf{x}^*_r)^\top$ from $\mathbf{X}_{n\times d}$ of size $r$
      with with sampling probability vector $\boldsymbol{\pi}$;
init
    Initialize cluster centroids $\mathbf{b}_1^{*(0)}, ..., \mathbf{b}_k^{*(0)}$ randomly;
repeat
    In iteration $t$, do the following steps:
    For each $\mathbf{x}^*_j$, $j = 1, 2, ..., r$, set label
      $y_j^{(t)} := \text{argmin}_{1\le l\le k} \frac{1}{n\pi_j^*}||\mathbf{x}^*_j - \mathbf{b}_l^{*(t-1)}||$;
    For each $l$, set $\mathbf{b}_l^{*(t)} := \frac{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\} * \frac{\mathbf{x}_j^*}{n\pi_j^*}}{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\}}$;
until Convergence criterion is met;
Output values from last iteration $t_l$: $\mathbf{b}^* = (\mathbf{b}_1^{*(t_l)\top}, ..., \mathbf{b}_k^{*(t_l)\top})^\top$
  and label $\mathbf{y} = (y_1^{(t_l)}, ..., y_n^{(t_l)})^\top$, where
  $y_i^{(t_l)} = \text{argmin}_{1\le l\le k} ||\mathbf{x}_i - \mathbf{b}_l^{*(t_l)}||^2$, $i = 1, 2, ..., n$
end
```

Remark 2.2.1 (Assumptions on π) *The assumptions for different weights may be different. Chatterjee, et al (2005) gave the assumptions of weights for the generalized bootstrap for estimating equations: π exchangeable, all π have the same expectation $E(\frac{1}{\pi_i}) = 1$, and the same finite variance $Var(\pi_i) = \sigma^2 < \infty$. For $W_i = (\pi_i - 1)/\sigma_n$, need $\sigma_n^2 = o(n)$, $E(W_iW_j) = O(\frac{1}{n})$ and $E(W_i^2W_j^2) \to 1$ for $i \neq j$, and $E(W_i^4) < \infty$.*

In our case, the assumptions are different from Chatterjee's in three parts: 1. π_i's are not exchangeable. 2. $Var(\pi_i)$ are not necessarily all equal for different i. 3. π_i's are data driven, therefore not independent with $\mathbf{X}$, since it contains information from data.

2.3 Theorem of Consistency

MacQueen (1967) proved weak consistency of k-means algorithm. David Pollard (1982) proved the strong consistency results. We will follow their ideas and prove the consistency of k-means algorithm via A-optimal subsampling. We shall continue using their notations.

As aforementioned, observations $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathbb{R}^d$ are made on probability distribution P. The corresponding empirical measure is denoted by P_n. Pollard defined in a more generalized case that for arbitrary (finite) subset A of $\mathbb{R}^d$ and arbitrary probability measure Q,

$$\Phi(A, Q) := \int \min_{\mathbf{a}\in A} \phi(||\mathbf{x}-\mathbf{a}||)Q(d\mathbf{x}), \tag{2.3.1}$$

and

$$m_k(Q) := \inf\{\Phi(A, Q) : \#\{A\} \geq k\}, \tag{2.3.2}$$

where ϕ is a positive non-decreasing function discussed in Remark(2.3.1), and $\#\{A\}$ denotes the cardinality of set A. The population version of Φ is

$$\Phi(A, P) = \int \min_{\mathbf{a}\in A} \phi(||\mathbf{x}-\mathbf{a}||)P(d\mathbf{x}). \tag{2.3.3}$$

This can be estimated by the empirical version,

$$\Phi(A, P_n) = \sum_{i=1}^{n} \frac{1}{n} \min_{\mathbf{a}\in A} \phi(||\mathbf{x}_i-\mathbf{a}||). \tag{2.3.4}$$

In addition, given k, $\bar{A} = \bar{A}(k)$ denotes the set of optimal cluster centroids for the population, and $A_n = A_n(k)$ denotes the set of optimal cluster centroids based on the sample. Thus by the above definitions, we have $\Phi(\bar{A}, P) = m_k(P)$ and $\Phi(A_n, P_n) = m_k(P_n)$.

Suppose $\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_r^*$ is a subsample taken from the full sample $\mathbf{x}_i, i = 1, 2, ..., n$ using sampling probabilities $\pi_1, \pi_2, ..., \pi_n$. Let $\pi_j^*,\ j = 1, 2, ..., r$ be the corresponding probabilities to the subsample. By putting probability mass $\frac{1}{rn\pi_1^*}, \frac{1}{rn\pi_2^*}, ..., \frac{1}{rn\pi_r^*}$ on each subsample data points, we construct Hansen-Hurwitz estimate $\Phi(A, \hat{P}_n)$ to approximate $\Phi(A, P_n)$ as follows:

$$\Phi(A, \hat{P}_n) = \sum_{j=1}^{r} \frac{1}{rn\pi_j^*} \min_{\mathbf{a}\in A} \phi(||\mathbf{x}_j^* - \mathbf{a}||). \tag{2.3.5}$$

This ensures the unbiasedness property:

$$E^*\Phi(A, \hat{P}_n) = \Phi(A, P_n).$$

Likewise,

$$m_k(\hat{P}_n) := \inf\{\Phi(A, \hat{P}_n) : \#\{A\} \geq k\}. \tag{2.3.6}$$

Let $A_r^* = A_r^*(k)$ be the set of optimal cluster centroids based on the subsample, that is $\Phi(A_r^*, \hat{P}_n) = m_k(\hat{P}_n)$, and let $\mathbf{w} = (w_1, w_2, ..., w_n)^\top$ have a scaled Multinomial distribution with number of trails r and parameter vector $\boldsymbol{\pi}$, write $\mathbf{w} \sim sMult(\boldsymbol{\pi}, r)$, so that

$$P(w_1 = \frac{k_1}{r\pi_1}, w_2 = \frac{k_2}{r\pi_2}, ..., w_n = \frac{k_n}{r\pi_n}) = \frac{r!}{\prod_{i=1}^n k_i!} \prod_{i=1}^{n} \pi_i^{k_i}, k_i \geq 0, \quad \sum_{i=1}^{n} k_i = r. \tag{2.3.7}$$

Pollard (1982) proved the almost sure convergence of A_n to $\bar{A}$ in Hausdorff metric, i.e., $d_H(A_n, \bar{A}) \to 0$, a.s., where the Hausdorff metric measures how far two sets are from each other in a metric space. For two non-empty sets W and V, the Hausdorff distance is defined as

$$d_H(W, V) = \max\{\sup_{w\in W} \inf_{v\in V} d(W, V), \sup_{v\in V} \inf_{w\in W} d(W, V)\}.$$

Our goal is to show $d_H(A_r^*, A_n) \to 0$ almost surely as $r \to 0$. As pointed out by Pollard (1982), by arranging the labeling into a suitable case, almost sure convergence for individual cluster centroids can be obtained consequently.

Remark 2.3.1 *For $\phi(x) = x^2$, we have $\phi(||\mathbf{x}-\mathbf{a}||) = ||\mathbf{x}-\mathbf{a}||^2$, which is the criterion used in the Within Sum of Squares from the standard method of k-means. In more general cases, ϕ can be other functions. To make the proof rigorous, some assumptions were specified by Pollard. First, ϕ needs to be non-decreasing and continuous, and $\phi(0) = 0$. Second, there exists some constant $\lambda \in \mathbb{R}$ such that $\phi(2x) \leq \lambda\phi(x)$ for every $x \in \mathbb{R}^+$. Third, as $x \to \infty$, $\phi(x) \to \infty$. Throughout, these assumptions shall be assumed.*

Remark 2.3.2 *The reason we use $\Phi(A, \hat{P}_n) = \sum_{j=1}^{r} \frac{1}{rn\pi_j^*} \min_{\mathbf{a}\in A} \phi(||\mathbf{x}_j^* - \mathbf{a}||)$ instead of using $\Phi^* = \frac{1}{r}\sum_{j=1}^{r} \min_{\mathbf{a}\in A} \phi(||\mathbf{x}_j^* - \mathbf{a}||)$ is that, $\Phi(A, \hat{P}_n)$ is a Hansen-Hurwitz estimator, and that the probability masses on the subsample are not $\frac{1}{n}$ but $\frac{1}{rn\pi_j^*}$, $j = 1, 2, ..., r$. One property of the Hansen-Hurwitz estimator is that it is unbiased: $E^*(\Phi(A, \hat{P}_n)) = \Phi(A, P_n)$. Clearly, $E^*(\Phi^*) \neq \Phi(A, P_n)$ except in the special case of the uniform sampling, when all π_i are equal to $\frac{1}{n}$. Hansen-Hurwitz estimator is a generally used estimator and technique in weighted resampling problems.*

Remark 2.3.3 *The difference of our proof and that of Pollard's is that we work on the subsample. Also it is worthwhile to note that, for subsampling, $r << n$. As $r \to \infty$, n will be forced to go to infinity.*

Theorem 2.3.1 (The Uniform SLLN) *Suppose $B(5M)$ is a closed ball with radius $5M$, centered at origin, $\mathcal{E}_k := \{A \subset B(5M) : \#\{A\} \leq k\}$. For $A \in \mathcal{E}_k$, let $g_A(x) := \min_{a\in A} \phi(||\mathbf{x} - \mathbf{a}||)$ be a P-integrable function on $\mathbb{R}^d$, and let $\mathcal{G}$ be the family of functions of the form $g_A(x)$. Then,*

$$\sup_{g\in\mathcal{G}} \left| \int g d\hat{P}_n - \int g dP_n \right| \to 0, \quad a.s. \tag{2.3.8}$$

Proof Following Pollard (1982), we will prove a sufficient condition for (2.3.8): for each $\epsilon > 0$, there exists a finite class $\mathcal{G}_\epsilon$ that, to each $g \in \mathcal{G}$, we can find functions $\bar{g}, \underline{g}$ such that $\bar{g} \leq g \leq \underline{g}$ and $\int(\bar{g} - \underline{g})dP_n < \epsilon$. To prove the sufficient condition, we apply the SLLN to each function g in the countable class $\mathcal{G}_{1/2} \cup \mathcal{G}_{1/3} \cup \mathcal{G}_{1/4}...$ with bound given below for $|\int gd\hat{P}_n - \int gdP_n|$,

$$\int (\bar{g} - \underline{g})dP_n + \max\{|\int \bar{g}d\hat{P}_n - \int \bar{g}dP_n|, |\int \underline{g}d\hat{P}_n - \int \underline{g}dP_n|\}.$$

In fact, all components of the above bound will converge to 0 as $r \to \infty$. The second term $\max\{|\int \bar{g}d\hat{P}_n - \int \bar{g}dP_n|, |\int \underline{g}d\hat{P}_n - \int \underline{g}dP_n|\}$ will converge to 0 by the SLLN. The first term will be proved later.

To find a $\mathcal{G}_\epsilon$ satisfies the condition above, we will need a δ-net for $B(5M)$, D_δ, which is a finite subset of $B(5M)$ that every single element of $B(5M)$ is of a distance that is not longer than δ, with at least one point of D_δ. The value of δ will need to satisfy some condition given later. Let $\mathcal{E}_{k,\delta} = \{A \in \mathcal{E}_k; A \subseteq D_\delta\}$. Then for $A' \in \mathcal{E}_{k,\delta}$, let $\mathcal{G}_\epsilon$ be the class of the functions in the form:

$$\min_{\mathbf{a} \in A'} \phi(||\mathbf{x} - \mathbf{a}|| + \delta) \;\text{ or }\; \min_{\mathbf{a} \in A'} \phi(||\mathbf{x} - \mathbf{a}|| - \delta)$$

for any $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_k\} \in \mathcal{E}_k$, by the definition of D_δ, there exists $A' = \{\mathbf{a}'_1, \mathbf{a}'_2, ..., \mathbf{a}'_k\} \subseteq D_\delta \in \mathcal{E}_{k,\delta}$ such that $d_H(A, A') < \delta$. Now write

$$\bar{g}_A := \min_{\mathbf{a} \in A'} \phi(||\mathbf{x} - \mathbf{a}|| + \delta) \;\text{ and }\; \underline{g}_A := \min_{\mathbf{a} \in A'} \phi(||\mathbf{x} - \mathbf{a}|| - \delta)$$

in which define

$$\phi(x) = \begin{cases} \phi(x), x \geq 0 \\ 0, \quad\; x < 0 \end{cases}$$

Since $||\mathbf{a}_i - \mathbf{a}'_i|| < \delta$ for $i = 1, 2, ..., k$, we have

$$||\mathbf{x} - \mathbf{a}_i|| = ||\mathbf{x} - \mathbf{a}'_i + \mathbf{a}'_i - \mathbf{a}_i|| \leq ||\mathbf{x} - \mathbf{a}'_i|| + ||\mathbf{a}'_i - \mathbf{a}_i|| \leq ||\mathbf{x} - \mathbf{a}'_i|| + \delta$$

$$||\mathbf{x} - \mathbf{a}_i|| = ||\mathbf{x} - \mathbf{a}'_i + \mathbf{a}'_i - \mathbf{a}_i|| \geq ||\mathbf{x} - \mathbf{a}'_i|| - ||\mathbf{a}'_i - \mathbf{a}_i|| \geq ||\mathbf{x} - \mathbf{a}'_i|| - \delta$$

for $i = 1, 2, ..., k$ and each $x \in \mathbb{R}^d$. And since ϕ is a non-decreasing function,

$$\min_{\mathbf{a}' \in A'} \phi(||\mathbf{x} - \mathbf{a}'|| - \delta) \leq \min_{a \in A} \phi(||\mathbf{x} - \mathbf{a}||) \leq \min_{\mathbf{a}' \in A'} \phi(||\mathbf{x} - \mathbf{a}'|| + \delta)$$

which indicates $\underline{g}_A \leq g_A \leq \bar{g}_A$. Then for $R > 5M + \delta$,

$$\int (\bar{g}_A(x) - \underline{g}_A)P_n(d\mathbf{x})$$

$$
\begin{aligned}
&= \int \min_{\mathbf{a} \in A'} [\phi(||\mathbf{x} - \mathbf{a}|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}|| - \delta)] P_n(d\mathbf{x}) \\
&\leq \int \sum_{i=1}^{k} [\phi(||\mathbf{x} - \mathbf{a}'_i|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}'_i|| - \delta)] P_n(d\mathbf{x}) \\
&= \int_{||\mathbf{X}|| \leq R} \sum_{i=1}^{k} [\phi(||\mathbf{x} - \mathbf{a}'_i|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}'_i|| - \delta)] P_n(d\mathbf{x}) \\
&\quad + \int_{||\mathbf{X}|| \geq R} \sum_{i=1}^{k} [\phi(||\mathbf{x} - \mathbf{a}'_i|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}'_i|| - \delta)] P_n(d\mathbf{x}) \\
&\leq k \sup_{||\mathbf{X}|| \leq R,} \sup_{\mathbf{a} \in B(5M)} |\phi(||\mathbf{x} - \mathbf{a}|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}|| - \delta)]| \\
&\quad + k \int_{||\mathbf{X}|| \geq R} [\phi(||\mathbf{x}|| + ||\mathbf{x}||)] P_n(d\mathbf{x}) \\
&\leq k \sup_{||\mathbf{X}|| \leq R,} \sup_{\mathbf{a} \in B(5M)} |\phi(||\mathbf{x} - \mathbf{a}|| + \delta) - \phi(||\mathbf{x} - \mathbf{a}|| - \delta)]| \\
&\quad + k \int_{||\mathbf{X}|| \geq R} \lambda \phi(||\mathbf{x}||) d(P_n - P) + k \int_{||\mathbf{X}|| \geq R} \lambda \phi(||\mathbf{x}||) P(d\mathbf{x})
\end{aligned}
$$

(For the third to fourth step of the above inequality, since when $||\mathbf{x}|| \geq R > 5M + \delta$ we have $||\mathbf{x}|| - \delta > 5M \geq ||\mathbf{a}'_i||$, so $||\mathbf{x}|| > ||\mathbf{a}'_i|| + \delta$, for $\mathbf{a}'_i \in A'$, $i = 1, 2, .., k$.) The first term in the last step can be made smaller than $\epsilon/3$ by finding a small enough δ (this is how δ value is selected) because of the uniform continuity of ϕ on $B(5M)$. The second will be less than $\epsilon/3$ for sufficiently large n by SLLN. The third term will be smaller than $\epsilon/3$ if R is large enough. The proof is complete. ■

Applying Theorem 2.3.1, we prove the following consistency result.

Theorem 2.3.2 (Consistency) *Assume that $\int \phi(||\mathbf{x}||) P(d\mathbf{x}) < \infty$ and that there exist unique set $\bar{A}(j)$ for which $\Phi(\bar{A}(j), P) = m_j(P) = inf\{\Phi(A, P) : \#\{A\} \leq j\}$ for each $j = 1, 2, ..., k$. Then $d_H(A^*_r, A_n) \to 0$ a.s., and $\Phi(A^*_r, \hat{P}_n) - m_k(P_n) \to 0$ a.s.*

Proof Following Pollard (1982), our proof consists of two stages: the first stage is to show A^*_r is included in some compact region of $\mathbb{R}^d$; the second stage is to show that $W(A^*_r, \hat{P}_n) - W(A_n, P_n)$ converges to zero uniformly over $\{A : \#A \leq k\} \subset B(5M)$, almost surely. To prove the first stage, the first step is to find an M large enough so that for the closed ball $B(M)$ with radius M and the origin as the center, there is at least one point of the set A_r^* that is contained in the $B(M)$ when n is large enough. In the second step, we will prove that all of the points of A_r^* are included in the ball $B(5M)$. Next, we will show that A_n is also included in this ball $B(5M)$. The second stage will be proved by applying the uniform SLLN Theorem (2.3.1). This will complete the proof.

Choose the ball K centered at the origin with radius r_0 which has positive P measure. Select a large enough M such that $\phi(M-r_0)P(K) > \int \phi(||\mathbf{x}||)P(d\mathbf{x})$. By the law of large number, we have, as $n \to \infty$,

$$\alpha_n := \phi(M-r_0)P_n(K) \to \alpha_0 := \phi(M-r_0)P(K) \quad a.s. \tag{2.3.9}$$

$$\beta_n := \int \phi(||\mathbf{x}||)P_n(d\mathbf{x}) \to \beta_0 := \int \phi(||\mathbf{x}||)P(d\mathbf{x}) \quad a.s. \tag{2.3.10}$$

By (2.3.9), there exist an N_1 that for $n > N_1$, α_n is in the neighborhood $\mathcal{N}_1 = (\alpha_0 - \frac{\alpha_0-\beta_0}{4}, \alpha_0 + \frac{\alpha_0-\beta_0}{4})$ of α_0. Also, by (2.3.10), we can also find a N_2 such that for $n > N_2$, β_n is in the neighborhood $\mathcal{N}_2 = (\beta_0 - \frac{\alpha_0-\beta_0}{4}, \beta_0 + \frac{\alpha_0-\beta_0}{4})$ of β_0. Therefore, when $n > \max(N_1, N_2)$, $\alpha_n > \frac{3\alpha_0+\beta_0}{4} > \frac{\alpha_0+\beta_0}{2} \geq \frac{\alpha_0+3\beta_0}{4} > \beta_n$.

By definition, $\Phi(A_r^*, \hat{P}_n) \leq \Phi(A_0, \hat{P}_n)$ holds for any set A_0 with $\#\{A_0\} \leq k$. Let $A_0 = \{0\}$, then as $r \to \infty$,

$$\Phi(A_0, \hat{P}_n) - \int \phi(||\mathbf{x}||)P_n(d\mathbf{x}) \to 0 \quad a.s. \tag{2.3.11}$$

This holds along almost all sample paths.

If for $n > \max(N_1, N_2)$ (infinitely many n), $B(M)$ does not contain any point of A_r^*, then

$$\begin{aligned}
\limsup_r \Phi(A_r^*, \hat{P}_n) &= \limsup_r \int \min_{\mathbf{a}\in A_r^*} \phi(||\mathbf{x}-\mathbf{a}||)\hat{P}_n(d\mathbf{x}) \\
&\geq \limsup_r \int_K \min_{\mathbf{a}\in A_r^*} \phi(||\mathbf{x}-\mathbf{a}||)\hat{P}_n(d\mathbf{x}) \\
&\geq \lim_r \int_K \phi(|M-r_0|)\hat{P}_n(d\mathbf{x}) \\
&= \lim_n \phi(|M-r_0|)P_n(K)
\end{aligned}$$

$$
\begin{aligned}
&= \lim_n \alpha_n \\
&\geq \frac{3\alpha_0 + \beta_0}{4} \\
&> \frac{\alpha_0 + \beta_0}{2} \\
&\geq \frac{\alpha_0 + 3\beta_0}{4} \\
&\geq \lim_n \beta_n \\
&= \lim_r \sup \Phi(A_0, \hat{P}_n)
\end{aligned}
$$

The second to the third step is illustrated by the graph below:

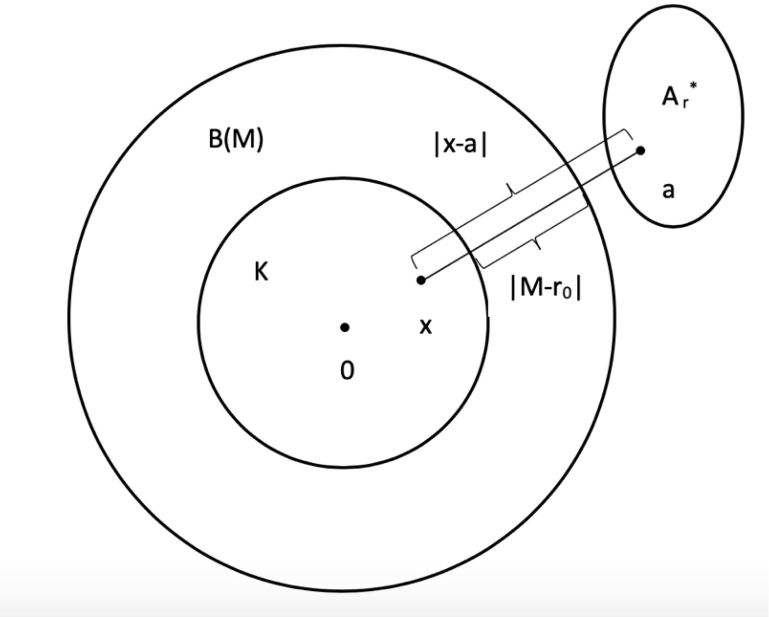


Figure 2.1.: Ball K and Ball $B(M)$

The inequality above makes $\Phi(A_r^*, \hat{P}_n) > \Phi(A_0, \hat{P}_n)$ happen infinitely often, which conflicts with the definition that $\Phi(A_r^*, \hat{P}_n)$ has the minimum value among all sets A of k or fewer points. Thus, without loss of generality, we can assume that there is at least one point of A_r^* which is contained in $B(M)$ almost surely.

Our second step is to show that all the points of A_r^* are contained in the ball $B(5M)$ for r and n large enough, where $B(5M)$ is the the ball centered at origin with radius $5M$. In this step, the case $k = 1$ has already been proved. For $k > 1$, we will prove by induction. Assume the result of the theorem holds for $1, 2, ..., k-1$ clusters, we will show that the results also holds for k. If, not all the points of A_r^* are contained in ball $B(5M)$ even for large r,, a contradiction will be shown as follows.

A second requirement for M is that, for $\epsilon > 0$ which satisfies

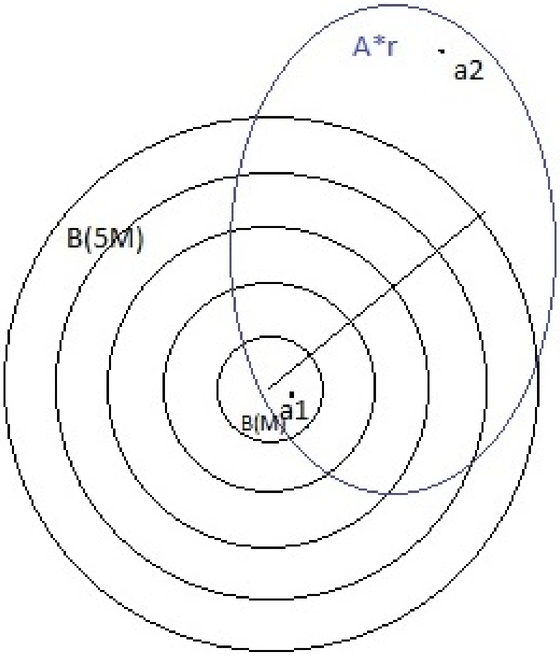
$$\epsilon + m_k(P) < m_{k-1}(P), \tag{2.3.12}$$

M needs to be so large that

$$\lambda \int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||) P(d\mathbf{x}) < \epsilon, \tag{2.3.13}$$

As we have proved in the first step, at least one point of A_r^* is in $B(M)$, let us name this point $\mathbf{a}_1$. Then $a_1 \in A_r^* \cap B(M)$. If we assume that A_r^* is not included in the ball $B(5M)$,then there is at least one point of A_r^* which is outside of the ball. If there is only one point, let us name this point $\mathbf{a}_2$, then $a_2 \in A_r^*/B(5M)$. The worse effect of deleting the point $\mathbf{a}_2$ from A_r^* is that all points that are in the cluster with centroid $\mathbf{a}_2$ are reassigned to the cluster with the centroid $\mathbf{a}_1$. These point are at least 2M from the origin, otherwise these points would have been originally assigned to $\mathbf{a}_1$ instead of $\mathbf{a}_2$. See figure (2.2).

Figure 2.2.: A_r^* and Ball $B(5M)$



If there are more than one centroids outside of $B(5M)$, the problem is similar. Then by deleting the centroid(s) outside $B(5M)$, $\Phi(\cdot, \hat{P}_n)$ will be increased by at most

$$\begin{aligned}\int_{||\mathbf{x}||\geq 2M} \phi(||\mathbf{x}-\mathbf{a}_1||)\hat{P}_n(d\mathbf{x}) &\leq \int_{||\mathbf{x}||\geq 2M} \phi(||\mathbf{x}|| + \phi||\mathbf{a}_1||)\hat{P}_n(d\mathbf{x}) \\ &\leq \int_{||\mathbf{x}||\geq 2M} \phi(2||\mathbf{x}||)\hat{P}_n(d\mathbf{x}) \\ &\leq \lambda \int_{||\mathbf{x}||\geq 2M} \phi(||\mathbf{x}||)\hat{P}_n(d\mathbf{x}).\end{aligned}$$

Assume $B_r^* = m_{k-1}(\hat{P}_n)$, which is the optimal set for $k-1$ clusters. Denote $\tilde{A}_r^* = A_r^*/B(5M)$, from which we can see $\tilde{A}_r^*$ is among the candidate sets of $k-1$ or fewer points that minimizes $\Phi(\cdot, \hat{P}_n)$. Therefore,

$$\Phi(\tilde{A}_r^*, \hat{P}_n) \geq \Phi(B_r^*, \hat{P}_n). \tag{2.3.14}$$

If there is a sub-sequence $A^*_{r_i}$ of $A^*_r \not\subset B(5M)$, then we have

$$
\begin{aligned}
0 &\leq \lim_i \left[\Phi(\tilde{A^*_{r_i}}, \hat{P}_n) - \Phi(B^*_r, \hat{P}_n)\right] \\
&\leq \overline{\lim_r}\big[\Phi(A^*_r, \hat{P}_n) + \lambda \int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||)\hat{P}_n(d\mathbf{x}) - \Phi(B^*_r, \hat{P}_n)\big] \\
&\leq \overline{\lim_r}\big[\Phi(A^*_r, \hat{P}_n) - \Phi(B^*_r, \hat{P}_n)\big] \\
&+ \lambda \overline{\lim_r}\big[\int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||)\hat{P}_n(d\mathbf{x}) - \int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||)P_n(d\mathbf{x})\big] \\
&+ \lambda \int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||)P(d\mathbf{x}) \\
&\leq \overline{\lim_r}\big[\Phi(A^*_r, \hat{P}_n) - \Phi(B^*_r, \hat{P}_n)\big] + \lambda \int_{||\mathbf{x}|| \geq 2M} \phi(||\mathbf{x}||)P(d\mathbf{x}) \\
&= \overline{\lim_r}\Phi(A, \hat{P}_n) - \Phi(\bar{B}, P) + \epsilon
\end{aligned}
$$

for any fixed A with $\#\{A\} \leq k$.

(Add some explanation of the above equation here)

Let $A = \bar{A}(k)$, then the above inequality will indicate that

$$
m_k(P) - m_{k-1}(P) + \epsilon \geq 0 \tag{2.3.15}
$$

This contradicts with (2.3.12). Therefore, A^*_r is included in $B(5M)$.

In the third step of the first stage, assume M is so large that the class of sets $\mathcal{E}_k := \{A \subset B(5M) : \#\{A\} \leq k\}$ contains $\bar{A}(k)$. Therefore the unique minimum of function $\Phi(\cdot, P)$ is achieved inside $B(5M)$ at $\bar{A}(k)$.

Since $B(5M)$ is compact as we we assumed, $\mathcal{E}_k$ is also a compact set under the topology induced by the Hausdorff metric. Pollard (1982) proved that the map $A \to \Phi(A, P)$ is continuous on $\mathcal{E}_k$, by definition and the uniform SLLN in Theorem 2.3.1, we have

$$
\Phi(A^*_r, \hat{P}_n) - \Phi(A_n, P_n) \leq \Phi(A_n, \hat{P}_n) - \Phi(A_n, P_n) \to 0, \quad a.s.
$$

On the other hand, we have

$$
\begin{aligned}
\Phi(A^*_r, \hat{P}_n) - \Phi(A_n, P_n) &= \Phi(A^*_r, \hat{P}_n) - \Phi(A^*_r, P_n) + \Phi(A^*_r, P_n) - \Phi(A_n, P_n) \\
&\geq \Phi(A^*_r, \hat{P}_n) - \Phi(A^*_r, P_n) + \Phi(A_n, P_n) - \Phi(A_n, P_n)
\end{aligned}
$$

$$= \Phi(A_r^*, \hat{P}_n) - \Phi(A_r^*, P_n) \to 0, \quad a.s.$$

Therefore, $\Phi(A_r^*, \hat{P}_n) - \Phi(A_n, P_n) \to 0$ almost surely as $r \to \infty$.

Now the rest is to prove that $d_H(A_r^*, A_n) \to 0$, a.s.. Since Pollard has already proved that $d_H(A_n, \bar{A}) \to 0$, a.s., A sufficient condition is that $d_H(A_r^*, \bar{A}) \to 0$, a.s.. In fact, if we can prove this condition, then

$$d_H(A_r^*, A_n) \leq d_H(A_r^*, \bar{A}) + d_H(A_n, \bar{A}) \to 0 \;\; a.s.$$

Suppose $d_H(A_r^*, \bar{A})$ does not converge to 0. Then either of the following cases will happen: (1). There exists a subsequence $A_{r_i}^*$ of A_r^* that diverges; (2). There exists a subsequence $A_{r_i}^*$ of A_r^* that converges to some fixed set $\bar{C}$ that is not equal to $\bar{A}$.

For the first case, the divergence of subsequence $A_{r_i}^*$ will lead to the divergence of A_r^*. However, since A_r^* is fully contained in the compact ball $B(5M)$, it must converge: a contradiction. For the second case, if $A_{r_i}^* \to \bar{C}$, a.s. then by the convergence result of $\Phi(A_r^*, \hat{P}_n)$ that we have proved,

$$\Phi(A_{r_i}^*, \hat{P}_n) \to \Phi(\bar{C}, P), \quad a.s.$$

since

$$\Phi(A_{r_i}^*, \hat{P}_n) \to \Phi(\bar{A}, P), \quad a.s.$$

we conclude

$$\bar{C} = \bar{A}, \quad a.s.$$

Thus we have obtained the contradiction. Therefore, the proof is now completed. ■

2.4 A Central Limit Theorem

2.4.1 Notation and Definitions

In this section, we will prove the central limit theorem for the optimal subsampling k-means cluster centroids. We introduce now the notations.

Let $\mathbf{I}_d$ denote the $d \times d$ matrix.

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)^\top$ be the full sample where $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, 2, ..., n$, and $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)^\top \in \mathbb{R}^n$ be the sampling distribution supported on the data points (assume $\boldsymbol{\pi}$ is known for now). Using $\boldsymbol{\pi}$, let $\mathbf{X}^* = (\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_r^*)^\top$ be the subsample drawn with replacement from $\mathbf{X}$ and $\boldsymbol{\pi}^* = (\pi_1^*, \pi_2^*, ..., \pi_r^*)^\top$ be the sampling probabilities corresponding to each point of $\mathbf{X}^*$.

Let $\boldsymbol{\mu}$ be a kd-dimension vector consisting of the true but unknown centroids $\boldsymbol{\mu}_s \in \mathbb{R}^d$ for $s = 1, 2, ..., k$. Let $\mathbf{b}_n$ be the kd-dimension vector of optimal k-means cluster centroids $\mathbf{b}_{ns} \in R^d$ for $s = 1, 2, ..., k$, based on sample $\mathbf{X}$, and let $\mathbf{b}^*$ be the kd-dimension vector of optimal k-means cluster centroids $\mathbf{b}_s^* \in R^d$ for $s = 1, 2, ..., k$, based on the subsample.

In addition, let $\mathbf{a}_n$ be a sequence of centroids approaching $\boldsymbol{\mu}$, and let $\mathbf{a}^*$ be a sequence approaching $\mathbf{b}_n$.

Let $\mathbf{M}_n = \{\mathbf{M}_{n1}, \mathbf{M}_{n2}, ..., \mathbf{M}_{nk}\}$ be the set of polyhedra associated with $\boldsymbol{\mu}$, let $\mathbf{B}_n = \{\mathbf{B}_{n1}, \mathbf{B}_{n2}, ..., \mathbf{B}_{nk}\}$ be the set of polyhedra associated with $\mathbf{b}_n$ and let $\mathbf{B}^* = \{\mathbf{B}_1^*, \mathbf{B}_2^*, ..., \mathbf{B}_k^*\}$ be the set of polyhedra associated with $\mathbf{b}^*$.

Let G_{st} denote the common face (possibly empty) of polyhedra $\mathbf{B}_{ns}$ and $\mathbf{B}_{nt}$, G_{st}^* denote the common face (possibly empty) of ployhedra $\mathbf{B}_s^*$ and $\mathbf{B}_t^*$, $s, t = 1, 2, ..., k$, $s \neq t$.

For $\mathbf{x} \in \mathbb{R}^d$ and vector $\mathbf{a} = (\mathbf{a}_1^\top, \mathbf{a}_2^\top, ..., \mathbf{a}_k^\top)^\top \in \mathbb{R}^{kd}$, let $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_k\}$ be the polyhedron associated with the centroid vector $\mathbf{a}$. Define

$$\phi(\mathbf{x}, \mathbf{a}) = \min_{1 \leq l \leq k} ||\mathbf{x} - \mathbf{a}_l||^2.$$

We shall use the notation throughout.

Let P be a probability measure, P_n be the empirical measure obtained by putting mass $\frac{1}{n}$ on each data point of the full sample $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$. For a measurable set A,

$$P_n(A) = \sum_{i=1}^{n} \frac{1}{n} \mathbf{1}[\mathbf{x}_i \in A].$$

For function $\phi(\mathbf{x}, \mathbf{a})$,

$$P_n\phi(\cdot, \mathbf{a}) = \sum_{i=1}^{n} \frac{1}{n} \phi(\mathbf{x}_i, \mathbf{a}).$$

The associated empirical process is

$$X_n(\cdot) = \sqrt{n}(P_n(\cdot) - P(\cdot)).$$

Let $\hat{P}_n$ be the empirical measure by placing mass $\frac{1}{rn\pi_1^*}, \frac{1}{rn\pi_2^*}, ..., \frac{1}{rn\pi_r^*}$ on each point of the subsample $\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_r^*$. for a measurable set A,

$$\hat{P}_n(A) = \sum_{j=1}^{r} \frac{1}{rn\pi_j^*} \mathbf{1}[\mathbf{x}_j^* \in A].$$

For function $\phi(\mathbf{x}, \mathbf{a})$,

$$\hat{P}_n \phi(\cdot, \mathbf{a}) = \sum_{j=1}^{r} \frac{1}{rn\pi_j^*} \phi(\mathbf{x}_j^*, \mathbf{a}).$$

The associated empirical process is

$$\hat{X}_n(\cdot) = \sqrt{r}(\hat{P}_n(\cdot) - P_n(\cdot)).$$

In fact, $\hat{P}_n \phi(\cdot, \mathbf{a})$ is a Hansen-Hurwitz estimate of $P_n \phi(\cdot, \mathbf{a})$, and clearly it is an unbiased estimator:

$$E^*(\hat{P}_n \phi(\cdot, \mathbf{a})) = P_n \phi(\cdot, \mathbf{a}).$$

By definition, $\mathbf{b}_n$ minimizes the within cluster sum of squares $W_n(\cdot)$, where

$$W_n(\mathbf{a}) = \frac{1}{n} \sum_{i=1}^{n} \min_{1 \le l \le k} ||\mathbf{x}_i - \mathbf{a}_l||^2.$$

$\mathbf{b}^*$ minimizes the within cluster sum of squares $\hat{W}_n(\cdot)$, and

$$\hat{W}_n(\mathbf{a}) = \frac{1}{r} \sum_{j=1}^{r} \frac{1}{n\pi_j^*} \min_{1 \le l \le k} ||\mathbf{x}_j^* - \mathbf{a}_l||^2.$$

Then the population within cluster sum of squares is

$$W(\mathbf{a}) = P\phi(\cdot, \mathbf{a}) = \int_{\mathbf{A}} \phi(\mathbf{x}, \mathbf{a}) P(d\mathbf{x}). \tag{2.4.1}$$

For the full sample,

$$W_n(\mathbf{a}) = P_n \phi(\cdot, \mathbf{a}) = P\phi(\cdot, \mathbf{a}) + \frac{1}{\sqrt{n}} X_n \phi(\cdot, \mathbf{a}). \tag{2.4.2}$$

For the subsample,

$$\hat{W}_n(\mathbf{a}) = \hat{P}_n \phi(\cdot, \mathbf{a}) = P_n \phi(\cdot, \mathbf{a}) + \frac{1}{\sqrt{r}} \hat{X}_n \phi(\cdot, \mathbf{a}). \tag{2.4.3}$$

Definition 2.4.1 (Quadratic mean differentiability) *Let $f(\mathbf{x},\mathbf{a})$ be defined on $\mathbb{R}^d \times \mathbb{R}^{kd}$, fix $\mathbf{x} \in \mathbb{R}^d$, $f(\mathbf{x},\mathbf{a})$ is differentiable in quadratic mean at $\mathbf{a}$ if*

$$E||f(\mathbf{x},\mathbf{a}+\mathbf{h}) - f(\mathbf{x},\mathbf{a}) - \dot{f}(\mathbf{x},\mathbf{a})^\top \mathbf{h}||^2 = o(||\mathbf{h}||^2), \quad as \ \ \mathbf{h} \to \mathbf{0}$$

2.4.2 Theorems

By reformulating the Lemma A of Pollard(1982), we have

Lemma 2.4.1 *Suppose $P||\mathbf{x}||^2 < \infty$, and P gives zero measure to every hyperplane in $\mathbb{R}^d$. Then the map $\mathbf{a} \to \phi(\mathbf{x},\mathbf{a}) \in L_2(P)$ is differentiable in quadratic mean in the sense*

$$E||\phi(\mathbf{x},\mathbf{a}+\mathbf{h}) - \phi(\mathbf{x},\mathbf{a}) - \dot{\phi}(\mathbf{x},\mathbf{a})^\top \mathbf{h}||^2 = o(||\mathbf{h}||^2).$$

And for every $\mathbf{x} \in R^d$, the derivative of $\phi(\mathbf{x},\mathbf{a})$ at $\mathbf{a}$ is

$$\dot{\phi}(\mathbf{x},\mathbf{a}) = \frac{\partial}{\partial \mathbf{a}}\phi(\mathbf{x},\mathbf{a}) = (-2(\mathbf{x}-\mathbf{a}_1)^\top \mathbf{1}[\mathbf{x} \in \mathbf{A}_1], ..., -2(\mathbf{x}-\mathbf{a}_k)^\top \mathbf{1}[\mathbf{x} \in \mathbf{A}_k])^\top.$$

Also, Hence, $E\phi(\mathbf{x},\mathbf{a})$ is differentiable with derivative

$$\begin{aligned}\boldsymbol{\gamma}(\mathbf{a}) &= \frac{\partial}{\partial \mathbf{a}} E\phi(\mathbf{x},\mathbf{a}) = E\frac{\partial}{\partial \mathbf{a}}\phi(\mathbf{x},\mathbf{a}) = E\dot{\phi}(\mathbf{x},\mathbf{a}) \\ &= (-2E(\mathbf{x}-\mathbf{a}_1)^\top \mathbf{1}[\mathbf{x} \in \mathbf{A}_1], ..., -2E(\mathbf{x}-\mathbf{a}_k)^\top \mathbf{1}[\mathbf{x} \in \mathbf{A}_k])^\top.\end{aligned}$$

Based on Lemma 2.4.1, we have

Proposition 2.4.1 *Suppose the conditions in Lemma 2.4.1 hold, then it holds in probability that $P_n\phi(\cdot,\mathbf{a})$ is differentiable in quadratic mean with derivative $P_n\dot{\phi}(\cdot,\mathbf{a})$ in the sense that the statement holds on an event whose probability (P) converges to 1 as n goes to infinity, where*

$$P_n\dot{\phi}(\cdot,\mathbf{a}) = (-2\sum_{i=1}^n (\mathbf{x}_i-\mathbf{a}_1)^\top \mathbf{1}[\mathbf{x}_i \in \mathbf{A}_1], ..., -2\sum_{i=1}^n (\mathbf{x}_i-\mathbf{a}_k)^\top \mathbf{1}[\mathbf{x}_i \in \mathbf{A}_k])^\top.$$

Proof By Pollard(1982), for $\mathbf{x} \in int\mathbf{A}_j$ and small enough $\mathbf{h}$,

$$\phi(\mathbf{x},\mathbf{a}+\mathbf{h}) = ||\mathbf{x}-\mathbf{a}_j-\mathbf{h}_j||^2 = \phi(\mathbf{x},\mathbf{a}) - 2\mathbf{h}_j^\top(\mathbf{x}-\mathbf{a}_j) + ||\mathbf{h}_j||^2,$$

Since the boundary of each $\mathbf{A}_j$, $j = 1, 2, ..., k$ has zero P_n measure, the function $\phi(\cdot, \mathbf{a}+\mathbf{h})$ can be expanded into

$$\phi(\mathbf{x}, \mathbf{a}+\mathbf{h}) = \phi(\mathbf{x}, \mathbf{a}) + \mathbf{h}^\top \dot{\phi}(\mathbf{x}, \mathbf{a}) + ||\mathbf{h}||R(\mathbf{x}, \mathbf{a}, \mathbf{h}), \quad for\ all\ \mathbf{x} \in \mathbb{R}^d \tag{2.4.4}$$

where

$$R(\mathbf{x}, \mathbf{a}, \mathbf{h}) \to 0 \;\; for\ almost\ all\ \mathbf{x}\ as\ \mathbf{h} \to \mathbf{0}.$$

Hence,

$$\begin{aligned} P_n\phi(\cdot, \mathbf{a}+\mathbf{h}) &= P_n\phi(\cdot, \mathbf{a}) + \mathbf{h}^\top P_n\dot{\phi}(\cdot, \mathbf{a}) + P_n||\mathbf{h}||R(\mathbf{x}, \mathbf{a}, \mathbf{h}) \\ &= P_n\phi(\cdot, \mathbf{a}) + \mathbf{h}^\top P_n\dot{\phi}(\cdot, \mathbf{a}) + ||\mathbf{h}||P_nR(\cdot, \mathbf{a}, \mathbf{h}). \end{aligned}$$

By simple algebra, we have

$$\begin{aligned} E||P_n\phi(\cdot, \mathbf{a}+\mathbf{h}) - P_n\phi(\cdot, \mathbf{a}) - \mathbf{h}^\top P_n\dot{\phi}(\cdot, \mathbf{a})||^2 &= ||\mathbf{h}||^2 E||\frac{1}{n}\sum_{i=1}^n R(\mathbf{x}_i, \mathbf{a}, \mathbf{h})||^2 \\ &\leq ||\mathbf{h}||^2 \frac{1}{n}\sum_{i=1}^n E||R(\mathbf{x}_i, \mathbf{a}, \mathbf{h})||^2 \\ &= ||\mathbf{h}||^2 E||R(\mathbf{x}, \mathbf{a}, \mathbf{h})||^2. \end{aligned}$$

Since $|R(\mathbf{x}, \mathbf{a}, \mathbf{h})| \in \mathcal{L}^2(P)$ by Pollard(1982), and $R(\mathbf{x}, \mathbf{a}, \mathbf{h}) \to 0$, a.s., we get $E||R(\mathbf{x}, \mathbf{a}, \mathbf{h})||^2 \to 0$ by Lebesgue's Dominated Convergence Theorem. Thus

$$E||P_n\phi(\cdot, \mathbf{a}+\mathbf{h}) - P_n\phi(\cdot, \mathbf{a}) - \mathbf{h}^\top P_n\dot{\phi}(\cdot, \mathbf{a})||^2 = o(||\mathbf{h}||^2).$$

Therefore, $P_n\phi(\cdot, \mathbf{a})$ is differentiable with derivative $P_n\dot{\phi}(\cdot, \mathbf{a})$ in probability.

■

Before giving the next Proposition, we introduce a few results first.

- Poissonization: suppose $N \sim Poisson(\lambda)$, and $(x_{N1}, x_{N2}, ..., x_{Nk}|N = n) \sim Mult(p_1, p_2, ..., p_k, n)$. Then $x_{N1}, x_{N2}, ..., x_{Nk}$ are independent with $x_{Ni} \sim Poisson(\lambda\pi_i)$;

- Packing number: Following Pollard (1990), we introduce the following. Define the packing number $D(\epsilon, T_0)$ for a subset T_0 of a metric space as the largest m such that there exist points $t_1, t_2, ..., t_m$ in T_0 with $d(t_i, t_j) < \epsilon$, for $i \neq j$;

- Envelope function: an envelope $\mathbf{F}$ is a vector such that $|f_i| \leq F_i$ for each $f_i \in \mathcal{F}$ and each i;

- Manageable: By Pollard (1990), we introduce the following. For each vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)$ of nonnegative constants, and each $\mathbf{f} \in \mathbb{R}^n$, define the pointwise product $\boldsymbol{\alpha} \bigodot \mathbf{f}$ to be the vector in $\mathbb{R}^n$ with the ith coordinate $\alpha_i \mathbf{f}_i$. Write $\boldsymbol{\alpha} \bigodot \mathcal{F}$ as the set of all vectors $\boldsymbol{\alpha} \bigodot \mathbf{f}$ with $\mathbf{f} \in \mathcal{F}$.

  Call a triangular array of processes $\{f_{ni}(w,t)\}$ manageable w.r.t the envelops $\mathbf{F}_n(w)$ if there exists a deterministic function λ such that

  (i) $\int_0^1 \sqrt{log\lambda(x)}dx < \infty$,

  (ii) $D(x|\boldsymbol{\alpha} \bigodot \mathcal{F}_n(w)|, \boldsymbol{\alpha} \bigodot \mathcal{F}_n(w)) \leq \lambda(x)$ for $0 < x \leq 1$, all w, all vectors α of non-negative weights and all n.

  Call a sequence of processes f_i manageable if the array defined by $f_{ni} = f_i$ for $i \leq n$ is manageable.

Proposition 2.4.2 *Let $\{\hat{\mathbf{a}}_n\}$ be an arbitrary sequence of random vectors in $\mathbb{R}^{kd}$ that satisfies $||\hat{\mathbf{a}}_n - \mathbf{b}_n|| = o_{p^*}(1)$, given vector $\mathbf{b}_n$ fixed. Assume conditions from Proposition (2.4.1) hold, and*

(i) $\sum_{i=1}^n \frac{2C^2}{n^2 r \pi_i^2}(1 + E||\mathbf{x}_i||^2) = O_p(1)$,

(ii) $\frac{\max_i \frac{\xi_{Ri}}{\pi_i}(1+||\mathbf{x}_i||)}{\sqrt{\sum_{i=1}^n \frac{\xi_{ri}^2}{\pi_i^2}(1+||\mathbf{x}_i||)^2}} = o_p(1)$*, where $R \sim Poisson(r)$ and $\xi_{R1}, ..., \xi_{Rn} \sim Poisson(1)$ and are independent.*

Then,

$$\hat{X}_n\phi(\cdot, \hat{\mathbf{a}}_n) = \hat{X}_n\phi(\cdot, \mathbf{b}_n) + (\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top \hat{X}_n\dot{\phi}(\cdot, \mathbf{b}_n) + \hat{\alpha}_B \tag{2.4.5}$$

where $\hat{\alpha}_B = o_{p^*}(||\hat{\mathbf{a}}_n - \mathbf{b}_n||)$.

Proof Firstly, apply $\hat{X}_n(\cdot)$ to equation (2.4.4) with $\mathbf{a} = \hat{\mathbf{a}}_n$ and $\mathbf{h} = \mathbf{h}_n := \hat{\mathbf{a}}_n - \mathbf{b}_n$ we have,

$$\hat{X}_n\phi(\cdot, \hat{\mathbf{a}}_n) - \hat{X}_n\phi(\cdot, \mathbf{b}_n) - (\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top \hat{X}_n\dot{\phi}(\cdot, \mathbf{b}_n)$$

$$
\begin{aligned}
&=||\hat{\mathbf{a}}_n - \mathbf{b}_n||\hat{X}_n R(\cdot, \mathbf{b}_n, \hat{\mathbf{a}}_n - \mathbf{b}_n)\\
&=||\mathbf{h}_n||\hat{X}_n R(\cdot, \mathbf{b}_n, \mathbf{h}_n).
\end{aligned}
$$

Now we need to prove that $\hat{X}_n R(\cdot, \mathbf{b}_n, \mathbf{h}_n) = o_{p^*}(1)$. By definition, $\hat{X}_n(\cdot) = \sqrt{r}(\hat{P}_n(\cdot) - P_n(\cdot))$, let $\bar{R}_n = P_n R(\cdot, \mathbf{b}_n, \mathbf{h}_n) = \frac{1}{n}\sum_{i=1}^n R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{h}_n)$, we have

$$
\begin{aligned}
\hat{X}_n R(\cdot, \mathbf{b}_n, \mathbf{h}_n) &= \sqrt{r}(\hat{P}_n - P_n) R(\cdot, \mathbf{b}_n, \mathbf{h}_n)\\
&= \sqrt{r}[\frac{1}{r}\sum_{j=1}^r \frac{R(\mathbf{x}_j^*, \mathbf{b}_n, \mathbf{h}_n)}{n\pi_j^*} - \bar{R}_n]\\
&= \sqrt{r}[\frac{1}{n}\sum_{j=1}^r \frac{R(\mathbf{x}_j^*, \mathbf{b}_n, \mathbf{h}_n)}{r\pi_j^*} - \bar{R}_n]\\
&\overset{d}{=} \sqrt{r}[\frac{1}{n}\sum_{i=1}^n [w_{ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{h}_n) - \bar{R}_n]
\end{aligned}
$$

where $(w_{r1}, w_{r2}, ..., w_{rn}) \sim sMult(\boldsymbol{\pi}, r)$, that is $P(W_{r1} = \frac{\xi_{r1}}{r\pi_1}, ..., W_{rn} = \frac{\xi_{rn}}{r\pi_n}) = \binom{r}{\xi_{r1}\xi_{r2}\cdots\xi_{rn}}\pi_1^{\xi_{r1}}\cdots\pi_n^{\xi_{rn}}$, $\xi_{r1} + \xi_{r2} + ... + \xi_{rn} = r$, $\xi_{r1} \geq 0, \xi_{r2} \geq 0, ..., \xi_{rn} \geq 0$. Then we can write

$$
\begin{aligned}
\hat{X}_n R(\cdot, \mathbf{b}_n, \mathbf{h}_n) &= \sqrt{r}\frac{1}{n}\sum_{i=1}^n [\frac{\xi_{Ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{h}_n)}{r\pi_i} - \bar{R}_n]\\
&= \sum_{i=1}^n [\frac{\xi_{Ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{h}_n)}{n\sqrt{r}\pi_i} - \frac{\sqrt{r}\bar{R}_n}{n}]
\end{aligned}
$$

By possionization, $\xi_{Ri} \sim Poisson(1)$, $i = 1, 2, ..., n$. Therefore $E(\xi_{Ri}^2) = (E(\xi_{Ri}))^2 + Var(\xi_{Ri}) = 2$.

Let $f_i(w, \mathbf{a})$ be a function of $\mathbf{a}$ for fixed $\mathbf{x}_i \in R^d$, $i = 1, 2, ..., n$, and $\mathcal{F}_n(w) = \{(f_1(w, \mathbf{a}), f_2(w, \mathbf{a}), ..., f_n(w, \mathbf{a})) : \mathbf{a} \in \mathcal{N}(\boldsymbol{\mu})\}$. Let $\mathbf{F}_n = (F_{n1}, F_{n2}, ..., F_{nn})^\top$ be the envelope function of $(f_1(w, \mathbf{a}), ..., f_n(w, \mathbf{a}))$, in which $\mathbf{F}_n$ will be specified later.

Pollard(1990) expanded the setting to cover triangular arrays of random processes,

$$
\{f_{ni}(w, t) : t \in T, 1 \leq i \leq n\} \ \ for\ n = 1, 2, ...,
$$

independent within each row. In our case

$$
f_{ni}(w, \mathbf{a}) = \frac{\xi_{Ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{a} - \mathbf{b}_n)}{n\sqrt{r}\pi_i}.
$$

Pollard(1982) proved that for $R(\mathbf{x}_i, \boldsymbol{\mu}, \mathbf{a} - \boldsymbol{\mu})$, the deterministic function $\lambda(x) = A(1/x)^W$ for some constants A and W. Hence, $R(\mathbf{x}, \boldsymbol{\mu}, \mathbf{a} - \boldsymbol{\mu})$ is Euclidean process and manageable with respect to the envelope function $\mathbf{F}_n$. For $\mathbf{b}_n$ and $\mathbf{a}$ in the neighbourhood of $\boldsymbol{\mu}$, $R(\mathbf{x}, \mathbf{b}_n, \mathbf{a} - \mathbf{b}_n)$ is also manageable (the packing number is also bounded by $\lambda(x)$ in the neighbourhood).

Next we specify the envelope function $\mathbf{F}_n$ from the inequality below. By Pollard(1982), $|R(\mathbf{x}, \mathbf{b}_n, \mathbf{h})| \leq C(1 + ||x||)$ for some constant C and $\mathbf{h}$ small enough, we have

$$\begin{aligned}|f_{ni}(w, \mathbf{a})| &= |\frac{\xi_{Ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{a} - \mathbf{b}_n)}{n\sqrt{r}\pi_i}| \\ &= \frac{\xi_{Ri}}{n\sqrt{r}\pi_i}|R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{a} - \mathbf{b}_n)| \\ &\leq \frac{\xi_{Ri}}{n\sqrt{r}\pi_i} C(1 + ||\mathbf{x}_i||) \\ &= F_{ni}\end{aligned}$$

Then we get,

$$\begin{aligned}P||\mathbf{F}_n||^2 &= E[\sum_{i=1}^{n} F_{ni}^2] \\ &= \sum_{i=1}^{n} E[\frac{\xi_{ri}^2}{n^2 r \pi_i^2} C^2 (1 + ||\mathbf{x}_i||)^2] \\ &= \sum_{i=1}^{n} \frac{C^2 E(\xi_{ri}^2)}{n^2 r \pi_i^2} E[(1 + ||\mathbf{x}_i||)^2] \\ &\leq \sum_{i=1}^{n} \frac{2C^2}{n^2 r \pi_i^2}(1 + E||\mathbf{x}_i||^2) \\ &= O_p(1)\end{aligned}$$

Now we consider the function

$$\Lambda_n(\frac{\delta_n}{||\mathbf{F}_n||}) = \int_0^{\frac{\delta_n}{||\mathbf{F}_n||}} \sqrt{log\lambda(x)}dx, \quad \delta_n = \sup_{\mathcal{F}_n(w)} |\mathbf{f}|,$$

where

$$\frac{\delta_n}{||\mathbf{F}_n||} = \frac{\max_i |f_{ni}|}{\sqrt{\sum_{i=1}^{n} F_{ni}^2}}$$

$$
\begin{aligned}
&= \frac{\max_i \frac{\xi_{Ri} R(\mathbf{x}_i, \mathbf{b}_n, \mathbf{a}-\mathbf{b}_n)}{n\sqrt{r}\pi_i}}{\sqrt{\sum_{i=1}^n \frac{\xi_{ri}^2}{n^2 r \pi_i^2} C^2(1+||\mathbf{x}_i||)^2}} \\
&\leq \frac{\max_i \frac{\xi_{Ri}}{n\sqrt{r}\pi_i} C(1+||\mathbf{x}_i||)}{\sqrt{\sum_{i=1}^n \frac{\xi_{ri}^2}{n^2 r \pi_i^2} C^2(1+||\mathbf{x}_i||)^2}} \\
&= o_p(1).
\end{aligned}
$$

Then $\Lambda_n(\frac{\delta_n}{||\mathbf{F}_n||})$ will converge to 0. Now we provide a sufficient condition for the last step of the above equation to hold: $E||\mathbf{x}_1||^t < \infty$ for $t > 2$.

After truncation of the sampling probabilities π_i, $\frac{1}{\pi_i}$ is bounded by some constant d. So the last term is approximately equal to

$$
\frac{\frac{\xi_{Ri}}{\sqrt{n}}(1+||\mathbf{x}_i||)}{\sqrt{\frac{1}{n}\sum_{i=1}^n \xi_{ri}^2 (1+||\mathbf{x}_i||)^2}}.
$$

For $\forall\varepsilon > 0$,

$$
\begin{aligned}
&P(\max_i \xi_{Ri}||\mathbf{x}_i|| > \varepsilon\sqrt{n}) \\
=&1 - P(\xi_{Ri}||\mathbf{x}_i|| \leq \varepsilon\sqrt{n}, i = 1, ..., n) \\
=&1 - [P(\xi_{r1}||\mathbf{x}_1|| \leq \varepsilon\sqrt{n})]^n.
\end{aligned}
$$

Denote $\eta_n = [P(\xi_{r1}||\mathbf{x}_1|| \leq \varepsilon\sqrt{n})]^n$. Then,

$$
\begin{aligned}
|\log \eta_n| &= |n \log P(\xi_{r1}||\mathbf{x}_1|| \leq \varepsilon\sqrt{n})| \\
&\leq \frac{E(\xi_{r1}^t ||\mathbf{x}_1||^t)}{\varepsilon^t n^{t/2-1}} \\
&= \frac{E\xi_{r1}^t E||\mathbf{x}_1||^t}{\varepsilon^t n^{t/2-1}}
\end{aligned}
$$

Since $\xi_{r1} \sim Poisson(1)$, $E\xi_{r1}^t$ is bounded. Under the sufficient condition we specified previously, $|\log \eta_n| = o_p(1)$ as $n \to \infty$. Therefore, the assumption (ii) hold.

Lastly, we applying the maximal inequality by Pollard(1990, section 7, inequality (7.8)) for the case $p = 2$, there exists a constant C_2 such that

$$
\hat{P}_n \sup_{\hat{\mathbf{a}}_n} |\hat{X}_n R(\mathbf{x}_i, \mathbf{b}_n, \hat{\mathbf{a}}_n - \mathbf{b}_n)|^2
$$

$$
\begin{aligned}
&= \hat{P}_n \sup_{\hat{\mathbf{a}}_n} |\sum_{i=1}^{n} \mathbf{f}_i(w, \hat{\mathbf{a}}_n) - \sqrt{r}\bar{R}_n|^2 \\
&\leq (18C_2)^2 P||\mathbf{F}_n||^2 \Lambda_n(\delta_n/||\mathbf{F}_n||^2) \\
&= o_p(1)
\end{aligned}
$$

Proof complete. ■

Reformulating the Lemma C of Pollard(1982), we have

Lemma 2.4.2 *Suppose $P||\mathbf{x}||^2 < \infty$, and $P(\cdot)$ has a continuous density $f(\cdot)$ w.r.t d dimensional Lebesgue measure. Assume for $i, j = 1, ..., k$,*

$$E\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^\top \mathbf{1}[\mathbf{x} \in G_{ij}]\}$$

exists and is continuous in $\mathbf{m} \in \mathbb{R}^d$. Then, if $\mathbf{a}_1, ..., \mathbf{a}_k$ are distinct, then $E\phi(\mathbf{x}, \mathbf{a})$ has a second derivative $\Gamma = \frac{\partial^2}{\partial \mathbf{a}^\top \partial \mathbf{a}} E\phi(\mathbf{x}, \mathbf{a})$ made of $d \times d$ blocks:

$$
\Gamma_{ij} = \begin{cases} 2P(\mathbf{A}_s)\mathbf{I}_d - 2\sum_{\alpha \neq i} \lambda_{i\alpha}^{-1} E\{(\mathbf{x} - \mathbf{a}_i)(\mathbf{x} - \mathbf{a}_i)^\top \mathbf{1}[\mathbf{x} \in G_{i\alpha}]\} & for \quad i = j \\ -2\sum_{i \neq j} \lambda_{ij}^{-1} E\{(\mathbf{x} - \mathbf{a}_i)(\mathbf{x} - \mathbf{a}_j)^\top \mathbf{1}[\mathbf{x} \in G_{ij}]\} & for \quad i \neq j \end{cases}
$$

where $\lambda_{ij} = ||\mathbf{a}_i - \mathbf{a}_j||$.

By Lemma 2.4.2, we have the following proposition.

Proposition 2.4.3 *Assume conditions in Lemma 2.4.2 hold, then $P_n\phi(\cdot, \mathbf{a}) = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i, \mathbf{a})$ has second derivative $\Gamma_n = \frac{\partial^2}{\partial \mathbf{a}^\top \partial \mathbf{a}} P_n\phi(\cdot, \mathbf{a})$ made of $d \times d$ blocks:*

$$
\Gamma_{nij} = \begin{cases} 2P_n(\mathbf{A}_s)\mathbf{I}_d - 2\sum_{\alpha \neq i} \lambda_{i\alpha}^{-1} E_n\{(\mathbf{x} - \mathbf{a}_i)(\mathbf{x} - \mathbf{a}_i)^\top \mathbf{1}[\mathbf{x} \in G_{i\alpha}^*]\} & for \quad i = j \\ -2\sum_{i \neq j} \lambda_{ij}^{-1} E_n\{(\mathbf{x} - \mathbf{a}_i)(\mathbf{x} - \mathbf{a}_j)^\top \mathbf{1}[\mathbf{x} \in G_{ij}^*]\} & for \quad i \neq j \end{cases}
$$

where $\lambda_{ij} = ||\mathbf{a}_i - \mathbf{a}_j||$, and

$$P_n(\mathbf{A}_s) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[\mathbf{x}_i \in \mathbf{A}_s].$$

This can be proved in the same way of Proposition (2.4.1) and Lemma 2.4.2.

Proposition 2.4.4 *Suppose $\mathbf{b_n}$ minimizes $W_n(\cdot)$. Let $\{\hat{\mathbf{a}}_n\}$ be an arbitrary sequence of random vectors in $\mathbb{R}^{kd}$ that satisfies $||\hat{\mathbf{a}}_n - \mathbf{b}_n|| = o_{p^*}(1)$. Assume assumptions in Lemma 2.4.2 hold, and assume $\mathbf{Y}_j^* = \frac{\phi(\mathbf{x}_j^*, \mathbf{b}_n)}{n\pi_j^*} - P_n\dot{\phi}(\cdot, \mathbf{b}_n)$, $j = 1, 2, ..., r$ satisfies Lindeberg's condition: for $\forall\varepsilon > 0$,*

$$\sum_{i=1}^{n} \pi_i ||\mathbf{Y}_i||^2 \mathbf{1}[||\mathbf{Y}_i|| > \sqrt{r}\varepsilon] = o_p(1), \quad \text{as } r \to \infty.$$

Then,

$$\hat{W}_n(\hat{\mathbf{a}}_n) = \hat{W}_n(\mathbf{b}_n) - \frac{1}{\sqrt{r}}\hat{\mathbf{Z}}_n^\top(\hat{\mathbf{a}}_n - \mathbf{b}_n) + \frac{1}{2}(\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top\Gamma_n(\hat{\mathbf{a}}_n - \mathbf{b}_n) + \alpha_D^* \tag{2.4.6}$$

where $\alpha_D^* = o_{p^*}(\frac{\lambda^*}{\sqrt{r}}) + o_{p^*}(\lambda^{*2})$, and $\lambda^* = ||\hat{\mathbf{a}}_n - \mathbf{b}_n||$. Also, $\tilde{\mathbf{V}}^{-1/2}\hat{\mathbf{Z}}_n$ has an asymptotic distribution $N(\mathbf{0}, \mathbf{I}_{kd})$, in which $\tilde{\mathbf{V}}$ is given in the proof below.

Proof By Proposition (2.4.1) and Proposition (2.4.3), it holds in probability that,

$$\begin{aligned} P_n\phi(\cdot, \mathbf{a}^*) =& P_n\phi(\cdot, \mathbf{b}_n) + (\mathbf{a}^* - \mathbf{b}_n)^\top\boldsymbol{\gamma}(\mathbf{b}_n) \\ &+ \frac{1}{2}(\mathbf{a}^* - \mathbf{b}_n)^\top\Gamma_n(\mathbf{a}^* - \mathbf{b}_n) + o_{p^*}(||\mathbf{a}^* - \mathbf{b}_n||^2). \end{aligned}$$

where both $\mathbf{a}^*$ and b_n are in the neighbourhood of $\boldsymbol{\mu}$. Since $\mathbf{b}_n$ minimizes $W_n(\cdot)$, thus the first derivative $\boldsymbol{\gamma}(\mathbf{b}_n)$ vanishes. Substitute $\mathbf{a}^*$ with $\hat{\mathbf{a}}_n$, we get

$$P_n\phi(\cdot, \hat{\mathbf{a}}_n) = P_n\phi(\cdot, \mathbf{b}_n) + \frac{1}{2}(\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top\Gamma_n(\hat{\mathbf{a}}_n - \mathbf{b}_n) + o_{p^*}(\lambda^{*2}). \tag{2.4.7}$$

By Proposition (2.4.2),

$$\hat{X}_n\phi(\cdot, \hat{\mathbf{a}}_n) = \hat{X}_n\phi(\cdot, \mathbf{b}_n) + (\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top\hat{X}_n\dot{\phi}(\cdot, \mathbf{b}_n) + o_{p^*}(||\hat{\mathbf{a}}_n - \mathbf{b}_n||). \tag{2.4.8}$$

Plugging equations (2.4.7) and (2.4.8) to (2.4.3), and rearranging the terms gives

$$\begin{aligned} \hat{W}_n(\hat{\mathbf{a}}_n) =& \hat{W}_n(\mathbf{b}_n) + \frac{1}{\sqrt{r}}(\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top\hat{X}_n\dot{\phi}(\cdot, \mathbf{b}_n) + o_{p^*}(\frac{\lambda^*}{\sqrt{r}}) \\ &+ \frac{1}{2}(\hat{\mathbf{a}}_n - \mathbf{b}_n)^\top\Gamma_n(\hat{\mathbf{a}}_n - \mathbf{b}_n) + o_{p^*}(\lambda^{*2}) \end{aligned}$$

Now we need to show the asymptotic distribution of $\hat{\mathbf{Z}}_n = -\hat{X}_n\dot{\phi}(\cdot, b_n)$. By definition of $\hat{X}_n$,

$$\hat{\mathbf{Z}}_n = -\sqrt{r}[\hat{P}_n\dot{\phi}(\cdot, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)]$$

$$= -\sqrt{r}[\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)]$$

The expectation of $\hat{\mathbf{Z}}_n$ is

$$\begin{aligned}
E^*(\hat{\mathbf{Z}}_n) &= -\sqrt{r}[\frac{1}{r}\sum_{j=1}^{r}E^*(\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n)) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)] \\
&= -\sqrt{r}[\sum_{i=1}^{n}\frac{1}{n\pi_i}\dot{\phi}(\mathbf{x}_i, \mathbf{b}_n)\pi_i - P_n\dot{\phi}(\cdot, \mathbf{b}_n)] \\
&= -\sqrt{r}[\frac{1}{n}\sum_{i=1}^{n}\dot{\phi}(\mathbf{x}_i, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)] = \mathbf{0}
\end{aligned}$$

The variance co-variance matrix of $\hat{\mathbf{Z}}_n$ is

$$\begin{aligned}
\tilde{\mathbf{V}} :&= Var^*(\hat{\mathbf{Z}}_n) \\
&= E^*(\hat{\mathbf{Z}}_n^{\otimes 2}) - [E^*(\hat{\mathbf{Z}}_n)]^{\otimes 2} \\
&= rE^*([\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2}) \\
&= rE^*([\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n)][\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n)]^\top) \\
&\quad - 2rE^*(\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n)P_n\dot{\phi}(\cdot, \mathbf{b}_n)^\top) + r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} \\
&= rE^*(\frac{1}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_i^*, \mathbf{b}_n)\dot{\phi}^\top(\mathbf{x}_j^*, \mathbf{b}_n)) \\
&\quad - 2r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} + r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} \\
&= \frac{1}{r}\sum\sum_{i=j}E^*(\frac{1}{n^2\pi_i^{*2}}\dot{\phi}(\mathbf{x}_i^*, \mathbf{b}_n)\dot{\phi}^\top(\mathbf{x}_i^*, \mathbf{b}_n)) \\
&\quad + \frac{1}{r}\sum\sum_{i\neq j}E^*(\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_i^*, \mathbf{b}_n)\dot{\phi}^\top(\mathbf{x}_j^*, \mathbf{b}_n)) - r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} \\
&= \frac{1}{r}\frac{r}{n^2}E^*(\frac{1}{\pi_i^{*2}}\dot{\phi}(\mathbf{x}_i^*, \mathbf{b}_n)^{\otimes 2}) + \frac{1}{r}\frac{r(r-1)}{n^2}[E^*(\frac{1}{\pi_i^*}\dot{\phi}(\mathbf{x}_i^*, \mathbf{b}_n))]^{\otimes 2} \\
&\quad - r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{\pi_i^2}\dot{\phi}(\mathbf{x}_i, \mathbf{b}_n)^{\otimes 2}\pi_i + \frac{(r-1)}{n^2}[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2}
\end{aligned}$$

$$
\begin{aligned}
&- r[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2} \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{\pi_i}\dot{\phi}(\mathbf{x}_i, \mathbf{b}_n)^{\otimes 2} + \frac{r(n^2+1)-1}{n^2}[P_n\dot{\phi}(\cdot, \mathbf{b}_n)]^{\otimes 2}
\end{aligned}
$$

Again, since $\mathbf{b}_n$ minimizes $W_n(\cdot)$, thus the first derivative at $\mathbf{b}_n$, $P_n\dot{\phi}(\cdot, \mathbf{b}_n)$ vanishes. Thus the (s, t)th block of matrix $\tilde{\mathbf{V}}$ is

$$
\frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{\pi_i}\dot{\phi}(\mathbf{x}_i, \mathbf{b}_n)^{\otimes 2} = \frac{4}{n^2}\sum_{i=1}^{n}\frac{\mathbf{1}[\mathbf{x}_i \in \mathbf{B}_{ns}]\mathbf{1}[\mathbf{x}_i \in \mathbf{B}_{nt}](\mathbf{x}_i - \mathbf{b}_{ns})(\mathbf{x}_i - \mathbf{b}_{nt})^\top}{\pi_i}. \quad (2.4.9)
$$

Since x can only belong to one cluster, we have $\mathbf{1}[x \in \mathbf{B}_{ns}]\mathbf{1}[\mathbf{x} \in \mathbf{B}_{nt}] = 0$. Therefore (2.4.9) reduces to

$$
\tilde{\mathbf{V}}_s = \frac{4}{n^2}\sum_{i=1}^{n}\frac{\mathbf{1}[\mathbf{x}_i \in \mathbf{B}_{ns}](\mathbf{x}_i - \mathbf{b}_{ns})(\mathbf{x}_i - \mathbf{b}_{nt})^\top}{\pi_i} \quad (2.4.10)
$$

Since

$$
\begin{aligned}
\hat{\mathbf{Z}}_n &= -\sqrt{r}[\frac{1}{r}\sum_{j=1}^{r}\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)] \\
&= -\frac{1}{\sqrt{r}}\sum_{j=1}^{r}[\frac{1}{n\pi_j^*}\dot{\phi}(\mathbf{x}_j^*, \mathbf{b}_n) - P_n\dot{\phi}(\cdot, \mathbf{b}_n)] \\
&= -\frac{1}{\sqrt{r}}\mathbf{Y}_j^*,
\end{aligned}
$$

where $\mathbf{Y}_1^*, \mathbf{Y}_2^*, ..., \mathbf{Y}_r^*$ are conditionally i.i.d.. Under Lindeberg's condition, by central limit theorem it holds in probability that $\hat{\mathbf{Z}}_n$ is asymptotically normal with mean zero and variance co-variance matrix $\tilde{\mathbf{V}}$. ■

Theorem 2.4.1 (Central Limit Theorem) *Let $\mathbf{b}_n$ be the vector of optimal k-means cluster centroids for a random sample from a distribution P on $\mathbb{R}^d$. Let $\mathbf{b}^*$ be the vector of optimal k-means cluster centers from a subsample drawn using a sampling distribution $\pi_1, ..., \pi_n$ on the sample points. Suppose*

(i) the vector $\boldsymbol{\mu}$ which minimizes the population within cluster sum of squares $W(\cdot)$ is unique up to relabeling of its coordinates;

(ii) $E_p||\mathbf{x}||^2 < \infty$;

(iii) *P has continuous density f w.r.t. Lebesgue measure* λ *on* $\mathbb{R}^d$*;*

(iv) *for* $\forall \mathbf{x} \in \mathbb{R}^d$*, there exists a dominating function* $g(\mathbf{x})$ *such that* $f(\mathbf{x}) \leq g(||\mathbf{x}||)$*, and that* $r^d g(r)$ *is integrable w.r.t. Lebesgue measure on* $\mathbb{R}^+$*;*

(v) *the second derivative matrix* $\Gamma = \frac{\partial^2}{\partial \mathbf{a} \partial \mathbf{a}^\top} P\phi(\cdot, \mathbf{a})$ *evaluated at* $\mathbf{a} = \boldsymbol{\mu}$ *is positive definite;*

(vi) $\mathbf{Y}_j^* = \frac{\phi(\mathbf{x}_j^*, \mathbf{b}_n)}{n\pi_j^*} - P_n\dot{\phi}(\cdot, \mathbf{b}_n)$*,* $j = 1, 2, ..., r$ *satisfies Lindeberg's condition: for* $\forall \varepsilon > 0$*,*

$$\sum_{i=1}^{n} \pi_i ||\mathbf{Y}_i||^2 \mathbf{1}[||\mathbf{Y}_i|| > \sqrt{r}\varepsilon] = o_p(1), \quad \textit{as } r \to \infty;$$

(vii) $\sum_{i=1}^n \frac{2C^2}{n^2 r \pi_i^2}(1 + E||\mathbf{x}_i||^2) = O_p(1)$*;*

(viii) $\frac{\max_i \frac{\xi_{Ri}}{\pi_i}(1+||\mathbf{x}_i||)}{\sqrt{\sum_{i=1}^n \frac{\xi_{ri}^2}{\pi_i^2}(1+||\mathbf{x}_i||)^2}} = o_p(1)$*, where* $R \sim Poisson(r)$ *and* $\xi_{R1}, ..., \xi_{Rn} \sim Poisson(1)$ *and are independent.*

Then it holds in probability that, $\sqrt{r}\Gamma_n \tilde{\mathbf{V}}^{-1/2}(\mathbf{b}^* - \mathbf{b}_n) \Rightarrow N(\mathbf{0}, \mathbf{I}_{kd})$*, where* $\Gamma_n = \frac{\partial^2}{\partial \mathbf{a} \partial \mathbf{a}^\top} P_n\phi(\cdot, \mathbf{a})$ *evaluated at* $\mathbf{a} = \mathbf{b}_n$*, and* $\tilde{\mathbf{V}}$ *is the* $kd \times kd$ *diagonal matrix consists of the block metrices*

$$\tilde{\mathbf{V}}_s = \frac{4}{n}\sum_{j=1}^{n}(\mathbf{x}_j - \mathbf{b}_{ns})(\mathbf{x}_j - \mathbf{b}_{ns})^\top \mathbf{1}[\mathbf{x}_j \in \mathbf{B}_{ns}].$$

Here $\mathbf{B}_{ns}$ *is the set in* $\mathbb{R}^d$ *in which the points closer to* $\mathbf{b}_{ns}$ *than to other* $\mathbf{b}_{nt}$ *for* $s, t = 1, 2, ..., k$

Proof Conditions (i) and (ii) are the conditions for the consistency of $\mathbf{b}^*$. Pollard(1982) proved that under conditions (iii) and (iv), $\int(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top \mathbf{1}[\mathbf{x} \in F_{st}]d\mathbf{x}$ exists and depends on the location of the centroids continuously for each $s, t = 1, 2..., k$ and for each fixed $\mathbf{m} \in \mathbb{R}^d$. So the continuity assumption in Lemma 2.4.2 hold. Condition (v) will be used later. Condition (vi) is the Lindeberg condition from Proposition 2.4.4 to ensure the asymptotic normality of $\mathbf{Z}^*$. Conditions (vii) and (viii) are the moment assumption and poissonization assumption from Proposition 2.4.2. In fact, these two assumptions could be satisfied if $\mathbf{x}_i$'s have finite second moment, and $\frac{\xi_{Ri}}{\pi_i}(1+||\mathbf{x}_i||)$ are stochastically bounded, $i = 1, ..., n$.

Let $\lambda^* = ||\mathbf{b}^* - \mathbf{b}_n||$. By definition, $\mathbf{b}^*$ minimizes $\hat{W}_n(\cdot)$, thus,

$$\hat{W}_n(\mathbf{b}^*) \leq \hat{W}_n(\mathbf{b}_n),$$

Applying Proposition (2.4.4) with $\hat{\mathbf{a}}_n = \mathbf{b}^*$ we get,

$$\hat{W}_n(\mathbf{b}^*) = \hat{W}_n(\mathbf{b}_n) - \frac{1}{\sqrt{r}}\hat{\mathbf{Z}}_n^\top(\mathbf{b}^* - \mathbf{b}_n) + \frac{1}{2}(\mathbf{b}^* - \mathbf{b}_n)^\top \Gamma_n (\mathbf{b}^* - \mathbf{b}_n) + \alpha_D^*$$

Therefore,

$$-\frac{1}{\sqrt{r}}\hat{\mathbf{Z}}_n^\top(\mathbf{b}^* - \mathbf{b}_n) + \frac{1}{2}(\mathbf{b}^* - \mathbf{b}_n)^\top \Gamma_n (\mathbf{b}^* - \mathbf{b}_n) + \alpha_D^* \leq \hat{W}_n(\mathbf{b}^*) - \hat{W}_n(\mathbf{b}_n) \leq 0 \quad (2.4.11)$$

Since Γ is positive definite, by definition, for any vector $\mathbf{y}$ that $||\mathbf{y}|| > 0$, we have

$$\frac{\mathbf{y}^\top \Gamma \mathbf{y}}{||\mathbf{y}||^2} \geq \lambda_{\min}(\Gamma) > 0,$$

where $\lambda_{\min}(\Gamma)$ is the minimum eigenvalue of Γ. By strong law of large number and the consistency theorem (2.3.2), Γ_n converge to Γ. So there exist a number N that, for $n > N$, $\frac{\mathbf{y}^\top \Gamma_n \mathbf{y}}{||\mathbf{y}||^2}$ is in the neighbourhood of $\frac{\mathbf{y}^\top \Gamma_n \mathbf{y}}{||\mathbf{y}||^2}$, such that

$$\frac{\mathbf{y}^\top \Gamma_n \mathbf{y}}{||\mathbf{y}||^2} \geq \lambda_n = \frac{\lambda_{\min}(\Gamma)}{2} > 0,$$

Therefore, we can also get that $||\Gamma_n^{-1}||$ is bounded by a positive value. By Proposition (2.4.4), $\hat{\mathbf{Z}}_n$ converges in distribution, so $\hat{\mathbf{Z}}_n = O_{p^*}(1)$. And $\hat{\alpha}_D = o_{p^*}(\frac{\lambda^*}{\sqrt{r}}) + o_{p^*}(\lambda^{*2})$ Substitute the above terms in inequality (2.4.11),

$$o_{p^*}(\frac{\lambda^*}{\sqrt{r}}) + o_{p^*}(\frac{\lambda^*}{\sqrt{r}}) \geq \lambda_{\min}(\Gamma)\lambda^{*2}, \quad (2.4.12)$$

which forces

$$\lambda^* = o_{p^*}(\frac{1}{\sqrt{r}}).$$

Hence $\hat{\alpha}_D = o_{p^*}(\frac{1}{r})$. Set $\boldsymbol{\theta}^* = \sqrt{r}(\mathbf{b}^* - \mathbf{b}_n)$ and use simple algebra, we have

$$||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 = (\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n)^\top(\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n)$$

$$
\begin{aligned}
&= (\boldsymbol{\theta}^{*\top}\Gamma_n^{1/2\top} - \hat{\mathbf{Z}}_n^\top\Gamma_n^{-1/2\top})(\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n) \\
&= \boldsymbol{\theta}^{*\top}\Gamma_n\boldsymbol{\theta}^{*\top} - 2\hat{\mathbf{Z}}_n^\top\boldsymbol{\theta}^* + \hat{\mathbf{Z}}_n^\top\Gamma_n^{-1}\hat{\mathbf{Z}}_n.
\end{aligned}
$$

Because $||\Gamma_n^{-1}||$ is bounded, $||\frac{1}{\sqrt{r}}\Gamma_n^{-1}\hat{\mathbf{Z}}_n||=\frac{1}{\sqrt{r}}||\Gamma_n^{-1}||\cdot||\hat{\mathbf{Z}}_n||=\frac{1}{\sqrt{r}}O_{p^*}(1) = o_{p^*}(1)$. Therefore we apply Proposition (2.4.4) to $\hat{\mathbf{a}}_n = \mathbf{b}_n + \frac{1}{\sqrt{r}}\Gamma_n^{-1}\hat{\mathbf{Z}}_n$ and get

$$
\begin{aligned}
\hat{W}_n(\mathbf{b}_n + \frac{1}{\sqrt{r}}\Gamma_n^{-1}\hat{\mathbf{Z}}_n) &= \hat{W}_n(\mathbf{b}_n) - \frac{1}{r}\hat{\mathbf{Z}}_n^\top\Gamma_n^{-1}\hat{\mathbf{Z}}_n + \frac{1}{2r}\hat{\mathbf{Z}}_n^\top\Gamma_n^{-1\top}\Gamma_n\Gamma_n^{-1}\hat{\mathbf{Z}}_n \\
&\quad + o_{p^*}(\frac{1}{r}\hat{\mathbf{Z}}_n^\top\Gamma_n^{-\top}\Gamma_n^{-1}\hat{\mathbf{Z}}_n) \\
&= \hat{W}_n(\mathbf{b}_n) - \frac{1}{2r}\hat{\mathbf{Z}}_n^\top\Gamma_n^{-1}\hat{\mathbf{Z}}_n + \hat{\alpha}_D.
\end{aligned}
$$

Now apply Proposition (2.4.4) again with $\hat{\mathbf{a}}_n = \mathbf{b}^*$, we have

$$
\begin{aligned}
\hat{W}_n(\mathbf{b}^*) &= \hat{W}_n(\mathbf{b}_n) - \frac{1}{r}\hat{\mathbf{Z}}_n^\top\boldsymbol{\theta}^* + \frac{1}{2}\boldsymbol{\theta}^{*\top}\Gamma_n\boldsymbol{\theta}^* + \alpha_D^* \\
&= \hat{W}_n(\mathbf{b}_n) - \frac{1}{r}\hat{\mathbf{Z}}_n^\top\boldsymbol{\theta}^* + \frac{1}{2r}||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 - \frac{1}{2n}\hat{\mathbf{Z}}_n^\top\Gamma_n^{-1}\hat{\mathbf{Z}}_n + \alpha_D^* \\
&= \hat{W}_n(\mathbf{b}_n + \frac{1}{\sqrt{r}}\Gamma_n^{-1}\hat{\mathbf{Z}}_n) + \frac{1}{2r}||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 + \alpha_D^*.
\end{aligned}
$$

By definition, $\hat{W}_n(\mathbf{b}^*) \leq \hat{W}_n(\mathbf{b}_n + \frac{1}{\sqrt{r}}\Gamma_n^{-1}\hat{\mathbf{Z}}_n)$. Thus

$$
\frac{1}{2r}||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 + \alpha_D^* \leq 0,
$$

which forces

$$
\frac{1}{2r}||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 = \alpha_D^* = o_{p^*}(\frac{1}{r})
$$

or

$$
||\Gamma_n^{1/2}\boldsymbol{\theta}^* - \Gamma_n^{-1/2}\hat{\mathbf{Z}}_n||^2 = o_{p^*}(1),
$$

which leads to

$$
\sqrt{r}(\mathbf{b}^* - \mathbf{b}_n) = \boldsymbol{\theta}^* = \Gamma_n^{-1}\hat{\mathbf{Z}}_n + \Gamma_n^{-1/2}o_{p^*}(1)
$$

Thus, $\sqrt{r}\Gamma_n\tilde{\mathbf{V}}^{-1/2}(\mathbf{b}^* - \mathbf{b}_n)$ converges to normal distribution with mean zero and variance co-variance $kd \times kd$ identity matrix when $r \to 0$. This completes the proof.

■

2.5 Optimal Sampling Probabilities

Since the variance matrix of $\mathbf{b}^*$ is a function of $\boldsymbol{\pi}$. We seek to minimize the trace of the variance matrix $\Sigma(\boldsymbol{\pi}) = \Gamma_n^{-1}\tilde{\mathbf{V}}(\boldsymbol{\pi})\Gamma_n^{-1}$ to find the optimal π such that the subsample k-means cluster centroids $\mathbf{b}^*$ have the minimum variance, therefore a more sufficient estimator.

Theorem 2.5.1 *In the subsampling k-means algorithm, the optimal sampling probability $\boldsymbol{\pi}$ is given by*

$$\pi_i \propto ||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||, \quad i = 1, 2, ..., n, \tag{2.5.1}$$

where $\psi(\mathbf{B}_n(\mathbf{x} - \mathbf{b}_n)) = (\mathbf{1}[\mathbf{x} \in \mathbf{B}_{n1}](\mathbf{x} - \mathbf{b}_{n1})^\top, ..., \mathbf{1}[\mathbf{x} \in \mathbf{B}_{nk}](\mathbf{x} - \mathbf{b}_{nk})^\top)$

Proof The variance matrix $\Sigma(\boldsymbol{\pi})$ can be written in form of summation with function ψ,

$$\Sigma(\boldsymbol{\pi}) = \Gamma_n^{-1}\tilde{\mathbf{V}}(\boldsymbol{\pi})\Gamma_n^{-1} = \frac{4}{n^2}\sum_{i=1}^{n} \frac{\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))^\top\Gamma_n^{-\top}}{\pi_i} \tag{2.5.2}$$

Then the trace of the variance matrix can be expressed as:

$$\tau(\boldsymbol{\pi}) = \mathrm{Tr}(\Sigma(\boldsymbol{\pi})) = \frac{4}{n^2}\sum_{i=1}^{n} \frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||^2}{\pi_i} \tag{2.5.3}$$

To find the optimal $\boldsymbol{\pi}$ that minimizes the summation above, we apply the Lagrange multiplier method. Let λ be the Lagrange multiplier. Then in our case,

$$f(\pi_1, \pi_2, ..., \pi_n) = \frac{4}{n^2}\sum_{i=1}^{n} \frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||^2}{\pi_i},$$

$$g(\pi_1, \pi_2, ..., \pi_n) = \pi_1 + \pi_2 + ... + \pi_n.$$

The constraint is $\pi_1 + \pi_2 + ... + \pi_n = 1$. So ∇f is a n dimension vector of

$$-\frac{8}{n^2}\frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||^2}{\pi_i^2} \quad, i = 1, 2, ...n$$

and

$$\nabla g = (1, 1, ..., 1).$$

Now we need to solve the equations

$$\begin{cases} -\frac{8}{n^2}\frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_1-\mathbf{b}_n))||^2}{\pi_1^2} = \lambda \\ -\frac{8}{n^2}\frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_2-\mathbf{b}_n))||^2}{\pi_2^2} = \lambda \\ \ldots \\ -\frac{8}{n^2}\frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_n-\mathbf{b}_n))||^2}{\pi_n^2} = \lambda \\ \pi_1 + \pi_2 + \ldots + \pi_n = 1 \end{cases}$$

After some algebra, we get,

$$\pi_i = \frac{||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||}{\sum_{i=1}^n ||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||}, \quad i = 1, 2, ..., n \tag{2.5.4}$$

To make it simple, we can write

$$\pi_i \propto ||\Gamma_n^{-1}\psi(\mathbf{B}_n(\mathbf{x}_i - \mathbf{b}_n))||, \quad i = 1, 2, ..., n.$$

■

2.6 Optimal Scoring Method

Since $\mathbf{b}_n$ is contained in the expression of $\boldsymbol{\pi}$, to calculate sampling probabilities, we need to computer $\mathbf{b}_n$ first. However, $\mathbf{b}_n$ is unknown and it is what we want to estimate. Therefore we follow the A-optimal scoring method from Peng and Tan (2018) to calculate sampling probability vector $\boldsymbol{\pi}$. We will estimate $\mathbf{b}_n$ by taking a pre-subsample first. To be specific, take a pre-subsample $\mathbf{X}_0^*$ from the full sample $\mathbf{X}$ with uniform sampling probabilities and small pre-subsample size r_0. Then apply k-means algorithm to $\mathbf{X}_0^*$ and get a pre-subsample estimate $\mathbf{b}_0^*$ of $\mathbf{b}_n$. Denote the set of polyhedra associated with $\mathbf{b}_0^*$ as $\mathbf{B}_0^*$, then π_i is calculated by

$$\pi_i \propto ||\Gamma_n^{-1}\psi(\mathbf{B}_0^*(\mathbf{x}_i - \mathbf{b}_0^*))||, \quad i = 1, 2, ..., n.$$

Then the rest is just to apply Algorithm 4: k-means clustering algorithm via A-optimal subsampling. Using optimal scoring method, we have an updated version of the k-means clustering via A-optimal subsampling algorithm below. Note that, when performing the algorithm, we need to truncate the sampling probabilities to satisfy the assumptions. So we use the near A-optimal $\boldsymbol{\pi}$,

$$\pi_i^{tr} = \begin{cases} \pi_{([np_0])}, & if \pi_i \leq \pi_{([np_0])} \\ \pi_i, & if \pi_i > \pi_{([np_0])}, \end{cases}$$

where p_0 is the truncation proportion and $\pi_{([np_0])}$ is the $[np_0]$th sorted π_i, $i = 1, 2, ..., n$.

Algorithm 5: k-means Clustering Algorithm via Scoring Method Optimal Subsampling

```
Input  : Data $\mathbf{X}_{n\times d} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\top$, $\mathbf{x}_i \in \mathbb{R}^d$. Number of
         clusters $k$, subsample size $r$ and pre-sample size $r_0$
         in $\mathbb{Z}^+$;
Output: Centroid vector $\mathbf{b}^*$, cluster label $\mathbf{y}$;
pre-calculation
    Take a pre-subsample with sample size $r_0$,
    Calculate pre-subsample $k$-means centroids vector $\mathbf{b}_0^*$,
    Calculate sampling distribution
init
    Initialize cluster centroids $\mathbf{b}_1^{*(0)}, ..., \mathbf{b}_k^{*(0)}$ randomly;
repeat
    In iteration $t$, do the following steps:
    For each $\mathbf{x}_j^*$, $j = 1, 2, ..., r$, set label
      $y_j^{(t)} := \text{argmin}_{1\leq l\leq k} \frac{1}{n\pi_j^*}||\mathbf{x}_j^* - \mathbf{b}_l^{*(t-1)}||$;
    For each $l$, set $\mathbf{b}_l^{*(t)} := \frac{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\} * \frac{\mathbf{x}_j^*}{n\pi_j^*}}{\sum_{j=1}^r \mathbf{1}\{y_j^{(t)}=l\}}$;
until Convergence criterion is met;
Output values from last iteration $t_l$: $\mathbf{b}^* = (\mathbf{b}_1^{*(t_l)\top}, ..., \mathbf{b}_k^{*(t_l)\top})^\top$
  and label $\mathbf{y} = (y_1^{(t_l)}, ..., y_n^{(t_l)})^\top$, where
  $y_i^{(t_l)} = \text{argmin}_{1\leq l\leq k} ||\mathbf{x}_i - \mathbf{b}_l^{*(t_l)}||^2$, $i = 1, 2, ..., n$
end
```

3. BOOTSTRAPPING VIA A-OPTIMAL SUBSAMPLING

In this chapter, we will introduce bootstrapping and it's generalization for massive data via A-optimal subsampling. Theoretical results and algorithms are provided.

3.1 Bootstrap

Suppose data points $\mathbf{x} = (x_1, x_2, ..., x_n)^\top$ are observed independently. Let $\theta(\mathbf{x})$ be the statistic of interest. For example, $\theta(\mathbf{x})$ could be sample mean, median, bias, or variance. To estimate the sampling distribution of $\theta(\mathbf{x})$, traditionally, statisticians derive formulae and do statistical reference for those statistics. However, in some cases when the form of $\theta(\mathbf{x})$ is too complicated, the explicit form of the distribution of the statistics may be too difficult to derive, or may not even exist. In this case, bootstrap, as a numerical method is widely applied.

To be specific, take a bootstrap sample $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)^\top$ by sampling with replacement from the original sample $\mathbf{x}$, then repeat this sampling procedure B times. Calculate the statistics $\theta(\mathbf{x}^*)$ for each sample, then from $\theta(\mathbf{x_1^*}), \theta(\mathbf{x_2^*}), ..., \theta(\mathbf{x_B^*})$ we can get an empirical estimate of the distribution of $\theta(\mathbf{x})$. For example, to estimate the sampling distribution of sample mean $\bar{X}$.

3.1.1 Bootstrapping in linear Regression

Suppose in the linear regression model, $\mathbf{y} = (y_1, y_2, ..., y_n)^\top$ is the response vector and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)^\top$ is the design matrix or covariate matrix with full rank p. y_i and $\mathbf{x}_i$ satisfies

$$y_i = \boldsymbol{\beta}^\top \mathbf{x}_i + \varepsilon_i, i = 1, 2, ..., n. \tag{3.1.1}$$

where $\boldsymbol{\beta}$ is the regression coefficient parameter in $\mathbb{R}^p$, $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ are random errors with independent and identical distributions of mean 0 and positive finite variance σ^2.

The ordinary least square estimate of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$, which is a common estimate of regression coefficient in linear regression. Asymptotic result of $\hat{\boldsymbol{\beta}}$ is given below:

$$\mathbf{V}^{-1/2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3.1.2}$$

where

$$\mathbf{V} = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}, \tag{3.1.3}$$

and $\mathbf{V}$ can be approximated by the sandwitch estimator

$$\hat{\mathbf{V}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\text{Diag}(\hat{\varepsilon}^2)\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}. \tag{3.1.4}$$

To estimate $\boldsymbol{\beta}$ in linear regression model and its sampling distribution using bootstrapping, there are mainly two ways.

The first is paired bootstrap, sometimes also called empirical bootstrap. Take a resample with replacement from the sample pairs $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$ with probabilities $P(\mathbf{x}_j^{u*} = \mathbf{x}_i, y_j^{u*} = y_i) = \frac{1}{n}, i = 1, 2, ..., n$, one can get an i.i.d. bootstrap sample

$$(\mathbf{x}_1^{u*}, y_1^{u*}), (\mathbf{x}_2^{u*}, y_2^{u*}), ..., (\mathbf{x}_n^{u*}, y_n^{u*}).$$

where u on the superscript denotes the 'uniform' sampling. Repeatedly taking resamples B times, then B bootstrap samples

$$(\mathbf{X}_1^{u*}, \mathbf{y}_1^{u*}), (\mathbf{X}_2^{u*}, \mathbf{y}_2^{u*}), ..., (\mathbf{X}_B^{u*}, \mathbf{y}_B^{u*})$$

are obtained. For each bootstrap sample, calculate the OLSE $\hat{\boldsymbol{\beta}}_b^{u*} = (\mathbf{X}_b^{u*\top}\mathbf{X}_b^{u*})^{-1}\mathbf{X}_b^{u*\top}\mathbf{y}_b^{u*}$, $b = 1, 2, ..., B$. Now we have a bootstrap sampling distribution

$$\hat{\boldsymbol{\beta}}_1^{u*}, \hat{\boldsymbol{\beta}}_2^{u*}, ..., \hat{\boldsymbol{\beta}}_B^{u*}$$

of $\hat{\boldsymbol{\beta}}$.

Freeman(1981) gave the following asymptotic properties for the estimate obtained from paired bootstrap:

$$\mathbf{V}^{\mathbf{u}*-1/2}(\boldsymbol{\beta}^{u*} - \hat{\boldsymbol{\beta}}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3.1.5}$$

where

$$\mathbf{V}^{\mathbf{u}*} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathrm{Diag}(\hat{\varepsilon}^2) \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} + O_p(\frac{1}{n}) \tag{3.1.6}$$

Paired bootstrap may not work well when there are influential points in the observations, i.e., there exist $\mathbf{x}_i$'s that are far away from other observations. In the case when these points are not selected in the bootstrap sample, the estimation of $\boldsymbol{\beta}$ could be biased. Therefore, residual bootstrap, another way of bootstrapping the regression models was proposed. The residuals of the linear model (3.1.1) can be denoted by

$$e_i = y_i - \hat{y}_i = y_i - \hat{\boldsymbol{\beta}}^\top \mathbf{x}_i$$

In residual bootstrap, we take resample of the residuals $e_1, e_2, ..., e_n$ with probabilities $P(\hat{\varepsilon}_j^* = e_i) = \frac{1}{n}$, $i = 1, 2, ..., n$ and get

$$\hat{\varepsilon}_1^*, \hat{\varepsilon}_2^*, ..., \hat{\varepsilon}_n^*.$$

From this bootstrapped sample of residuals, we can construct the residual bootstrap samples

$$(\mathbf{x}_i^* = \mathbf{x}_i, y_i^* = \hat{\boldsymbol{\beta}}^\top \mathbf{x}_i + \hat{\varepsilon}_i^*), \quad i = 1, 2, ..., n.$$

For each sample, the residual bootstrap regression coefficient estimate can be calculated, and then we can use them to obtain the sampling distribution in a way similar to that of the paired bootstrapping.

Based on residual bootstrap, the wild bootstrap for linear regression models was proposed by Wu(1986). The difference between the two is that, the wild bootstrap does not take resample from the residuals. Instead, the wild bootstrap multiplies each residual with a normally distributed perturbation random variable on each residual to get the bootstrap residual. After that, the sampling distribution of parameter estimate is found in the same way.

In our research, we focus on the paired bootstrap (algorithm is given below) by generalizing the sampling probabilities $P(\mathbf{x}_j^{u*} = \mathbf{x}_i, y_j^{u*} = y_i)$ from uniform to non-uniform. In this case, the influential points could be selected with larger probability. In the future, we can generalize our work to the residual bootstrap and wild bootstrap.

Algorithm 6: Linear Regression Model Bootstrapping Algorithm

```
Input  : $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. $r,\ B \in \mathbb{Z}^{+}$.
Output: $\hat{\boldsymbol{\beta}}^{*} \in \mathbb{R}^{p\times 1}$
for $b$ in $1:B$ do
    Draw $r$ rows from $(\boldsymbol{X}, \boldsymbol{y})$ with replacement,
      obtain the bootstrap sample $(\boldsymbol{X}^{*}, \boldsymbol{y}^{*})$;
    For $(\boldsymbol{X}^{*}, \boldsymbol{y}^{*})$, calculate ordinary least squares
      estimate $\hat{\boldsymbol{\beta}}_{\boldsymbol{b}}^{*} = (\boldsymbol{X}^{*\top}\boldsymbol{X}^{*})^{-1}\boldsymbol{X}^{*\top}\boldsymbol{y}^{*}$;
end
Now use $\hat{\boldsymbol{\beta}}^{*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\boldsymbol{\beta}}_{b}^{*}$ to estimate $\hat{\boldsymbol{\beta}}$ and use $\hat{\boldsymbol{\beta}}_{b}^{*}$,
  $b = 1, 2, ..., B$ to estimate the sampling distribution
  of $\hat{\boldsymbol{\beta}}$.
end
```

3.1.2 Massive Data Bootstrapping

Bootstrapping is commonly used for small sample size. With the development of internet, more and more data generated nowadays have large sample size n, or large dimension of features p, or large product $n \times p$. In this case, the matrix multiplication and inverse matrix calculation could be time consuming or even not possible for regular computers. Methods like divide and conquer, subsampling, r out of n bootstrap, bags of little bootstraps were proposed. Divide and conquer can use parallel computing to save computing time, but it is hard to consider the association between different computing clusters. Subsampling takes subsamples without replacement. r out of n bootstrap takes bootstrap subsamples with replacement, and when r is smaller than n time is saved. Bags of little bootstrap was proposed to improve the robustness. However, the methods above do not include the data information in sampling probabilities.

Bootstrapping large samples have an advantage over bootstrapping small samples,As the sample size gets large, the higher order remainder terms in our theoretical results could be negligible, thus we can apply theoretical results in bootstrap to save time.

As mentioned before, uniform subsampling or bootstrapping is usually not the best way of extracting important information as they treat all observations with equal importance. Therefore, we calculate a sampling probability distribution before taking the bootstrap sample in order to make better use of the more informative observations hence improve efficiency of the estimation process.

3.2 Massive Data Bootstrapping via A-optimal Subsampling

In our work, we focus on improving the r out of n bootstrap, in which we choose the bootstrap subsample size $r << n$, and calculate the optimal sampling probability $\boldsymbol{\pi}$ which leads to an estimate with minimized MSE.

The idea can be considered as a weighted bootstrap with resample size $r << n$ and non-exchangeable data driven weights. To be specific, supposed a sampling distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)^\top$ on n data pairs $(\mathbf{x}_i, y_i)$, $i = 1, 2, ..., n$ is calculated before sampling. Take a bootstrap subsample of size $r << n$ from the pairs

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$$

with probability $P(\mathbf{x}_j^{o*} = \mathbf{x}_i, y_j^{o*} = y_i) = \pi_i, i = 1, 2, ..., n$, one can get an i.i.d. bootstrap sample

$$(\mathbf{x}_1^{o*}, y_1^{o*}), (\mathbf{x}_2^{o*}, y_2^{o*}), ..., (\mathbf{x}_m^{o*}, y_m^{o*}).$$

where o on the superscript denotes the 'optimal' sampling. Also a diagonal matrix $\mathbf{W}^* = \text{Diag}(\frac{1}{n\boldsymbol{\pi}^*})$ can be constructed, where $\boldsymbol{\pi}^*$ is the probability vector corresponds to the weighted bootstrap subsample. Calculate the r out of n bootstrap subsample weighted least square estimate $\hat{\boldsymbol{\beta}}^{o*} = (\mathbf{X}_b^{o*\top}\mathbf{W}^*\mathbf{X}_b^{o*})^{-1}\mathbf{X}_b^{o*\top}\mathbf{W}^*\mathbf{y}_b^{o*}$, which is a Hanson-Hurwitz estimator of $\hat{\boldsymbol{\beta}}$. We use this estimator because the sample is drawn with weight.

If $\pi_i = \frac{1}{n}$, $i = 1, 2, ..., n$, the above becomes the regular r out of n bootstrap, and $\hat{\boldsymbol{\beta}}^{o*}$ becomes $\hat{\boldsymbol{\beta}}^{u*}$.

The algorithm is given below.

Algorithm 7: Massive Data Linear Regression Model Bootstrapping via A-optimal Subsampling Algorithm

```
Input  : $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. $r,\ B \in \mathbb{Z}^{+}$.
Output: $\hat{\boldsymbol{\beta}}^{*} \in \mathbb{R}^{p\times 1}$
init
    Construct a sampling distribution $\boldsymbol{\pi} = (\pi_1, ..., \pi_n)$ for the input data$(\boldsymbol{X}, \boldsymbol{y})$;
for $b$ in $1 : B$ do
    Draw $r$ rows from $(\boldsymbol{X}, \boldsymbol{y})$ with replacement using the sampling distribution of $\boldsymbol{\pi}$ ;
    Formulate weight matrix $\boldsymbol{W}^* = \text{Diag}(1/r\boldsymbol{\pi}^*)$ of the resample $(\boldsymbol{X}^*, \boldsymbol{y}^*)$ with corresponding probabilities $\boldsymbol{\pi}^*$;
    Calculate the resample weighted least squares estimator $\hat{\boldsymbol{\beta}}^{*}_{r,b} = (\mathbf{X}^{*\top}\mathbf{W}^*\mathbf{X}^*)^{-\mathbf{1}}\mathbf{X}^{*\top}\mathbf{W}^*\mathbf{y}^*$;
end
Now use $\hat{\boldsymbol{\beta}}^{*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\boldsymbol{\beta}}^{*}_{b}$ to estimate $\hat{\boldsymbol{\beta}}$.
end
```

3.3 Asymptotic Theories

Firstly of all, the following conditions are introduced:

(M1) $\mathbf{x}_i$ and ε_i, $i = 1, 2, ..., n$ need to satisfy

$$\frac{1}{n^2}\sum_{i=1}^{n}\frac{\mathbf{x}_i\mathbf{x}_i^\top(\varepsilon_i^2-\sigma^2)}{\pi_{n,i}} = o(1), \quad a.s.$$

(M21) There exists a $p \times p$ matrix $\mathbf{M}_0$ which is positive definite, such that

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^\top = \mathbf{M}_0 + o(1).$$

(M22) There exist constants b and B such that, the minimum eigenvalues and maximum eigenvalues of the matrix $\mathbb{L}_n(\boldsymbol{\pi}_n) := \frac{1}{n^2}\sum_{i=1}^{n}\frac{\mathbf{x}_i\mathbf{x}_i^\top}{\pi_{n,i}}$ satisfy

$$0 < b \leq \lambda_{\min}(\mathbb{L}_n(\boldsymbol{\pi}_n))$$

and

$$\lambda_{\min}(\mathbb{L}_n(\boldsymbol{\pi}_n)) \leq B < \infty$$

(M3) The double array $\boldsymbol{\eta}_{n,i} := \mathbf{x}_i\varepsilon_i/n\pi_{n,i}$, $i = 1, 2, ..., n$, $n = 1, 2, ...$ satisfies the Lindeberg condition: for every $t > 0$,

$$\sum_{i=1}^{n}\pi_{n,i}||\boldsymbol{\eta}_{n,i}||^2\mathbf{1}[||\boldsymbol{\eta}_{n,i}|| \geq \sqrt{r}t] = o(1), \quad a.s. \;\; r \to \infty.$$

(M4) $\pi_{n,i}$ and $\mathbf{x}_i$ satisfy

$$\frac{1}{n^2}\sum_{i=1}^{n}\frac{||\mathbf{x}_i||^4}{\pi_{n,i}}, \quad n = 1, 2, ...$$

The assumptions above are moment assumptions and Lindeberg condition from Peng and Tan (2018), which are similar to the assumptions given by Zhu, *et al.* (2015). Below we present three theorems from Peng and Tan (2018) about the limiting property of $\hat{\boldsymbol{\beta}}^*$. Based on their theorems we give our theorem and proof at last.

Theorem 3.3.1 *Expand $\hat{\boldsymbol{\beta}}^*$ at the OLSE $\hat{\boldsymbol{\beta}}$, we have*

$$\hat{\boldsymbol{\beta}}^* = \hat{\boldsymbol{\beta}} + \frac{1}{r}\sum_{j=1}^{n}(\mathbf{X}^\top\mathbf{X})^{-1}\frac{x_j^*\hat{\varepsilon}_j^*}{\pi_j^*} + \mathbf{r}^*, \tag{3.3.1}$$

where $\mathbf{r}^*$ *is given by*

$$\mathbf{r}^* = \mathbf{r}^*(\hat{\varepsilon}^*) = ((\mathbf{X}^{*\top}\mathbf{W}^*\mathbf{X}^*)^{-\mathbf{1}} - (\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}})(\mathbf{X}^{*\top}\mathbf{W}^*\hat{\boldsymbol{\varepsilon}}^*). \tag{3.3.2}$$

Suppose (M1)-(M2) and (M4) hold. Then the remainder $\mathbf{r}^*$ *satisfies*

$$\mathbf{r}^* = O_{p^*}(\frac{1}{r}), a.s.$$

The empirical bias of $\hat{\boldsymbol{\beta}}^*$ *is*

$$Bias^*(\hat{\boldsymbol{\beta}}^*) = E^*(\hat{\boldsymbol{\beta}}^*) - \hat{\boldsymbol{\beta}} = -\frac{1}{r}\sum_{i=1}^{n}(\mathbf{X}^\top\mathbf{X})^{-1}\frac{h_{i,i}}{\pi_i}\mathbf{x}_i\hat{\varepsilon}_i^* + \mathbf{r}_1, \tag{3.3.3}$$

where $h_{i,i}$ *is the ith diagonal element of the hat matrix,* $i = 1, 2, ..., n$*. And* $\mathbf{r}_1 = O_p(\frac{1}{r^{3/2}})$*. Moreover, the variance co-variance matrix of* $\hat{\boldsymbol{\beta}}^*$ *can be expanded as*

$$\mathbf{V}^* = \frac{1}{r}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top Diag(\frac{\hat{\boldsymbol{\varepsilon}}^2}{\boldsymbol{\pi}})\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} + O_p(\frac{1}{r^2}) \tag{3.3.4}$$

Theorem (3.3.1) implies the following theorem,

Theorem 3.3.2 (Central Limit Theorem I) *Assume(M1)-(M4) hold and for all* $\varrho > 0$*,*

$$\max_{1\le i\le n} \|\boldsymbol{x}_i\| = o(n^{1/2}log^{-\varrho}(n)).$$

Assume there exists some $\rho > 2$ *such that*

$$\mathbf{E}(|\varepsilon_1|^\rho) < \infty.$$

Then $\hat{\boldsymbol{\beta}}^*$ *is asymptotically normal along almost all the sample path of the sequence* $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...,$

$$\sqrt{r}\mathbf{V}^{-\mathbf{1/2}}(\boldsymbol{\pi})(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}}) \Rightarrow \mathcal{N}(\mathbf{0}, \mathbf{I}_p), \quad a.s. \;\; r \to \infty \tag{3.3.5}$$

where

$$\mathbf{V}(\boldsymbol{\pi}) = (\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}\mathbf{X}^\top Diag(\frac{\hat{\boldsymbol{\varepsilon}}^2}{\boldsymbol{\pi}})\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}} \tag{3.3.6}$$

Based on the asymptotic normal distribution result, we can construct confidence of $\hat{\boldsymbol{\beta}}$, which will be discussed in the next section. Now we give the optimal sample probability in the following theorem.

Theorem 3.3.3 $\boldsymbol{\pi}^o$ *given below is the optimal sampling probability vector that minimizes* $\mathbf{V}(\boldsymbol{\pi})$,

$$\pi_i^o = \frac{||(\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}\mathbf{x_i}||}{\sum_{j=1}^n ||(\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}\mathbf{x_j}||}, \quad i = 1, 2, ..., n$$

The optimal sampling probability $\boldsymbol{\pi}$ is obtained from the method of Lagrange multiplier. The above results are for $(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta})$ since $\hat{\boldsymbol{\beta}}$ is unknown and what we are estimating in massive data linear regression model. However, the asymptotic results for $(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0)$ is also what we are interested in. Following Theorem (3.3.2), we focus on inference on the true parameter β_0 in massive data bootstrap and give the theorem below.

Theorem 3.3.4 (Central Limit Theorem II) *Assume the assumptions of Theorem (3.3.2) hold, and* $r = o(n)$, *then* $\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0$ *is asymptotically normal,*

$$\sqrt{r}\mathbf{V}^{-\mathbf{1/2}}(\boldsymbol{\pi})(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0) \Rightarrow \mathcal{N}(\mathbf{0}, \mathbf{I}_p), \quad a.s. \quad r \to \infty \tag{3.3.7}$$

where

$$\mathbf{V}(\boldsymbol{\pi}) = (\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}\mathbf{X}^\top Diag(\frac{\hat{\boldsymbol{\varepsilon}}^2}{\boldsymbol{\pi}})\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}} \tag{3.3.8}$$

Proof We can rewrite $\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0$ and get

$$\begin{aligned}\sqrt{r}(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0) &= \sqrt{r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \\ &= \sqrt{r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}}) + \sqrt{r}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \\ &= \sqrt{r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}}) + \frac{\sqrt{r}}{\sqrt{n}}\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)\end{aligned}$$

Since $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)$ converges in distribution, we have $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) = O_p(1)$. Apply the assumption $r = o(n)$,

$$\sqrt{r}(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}_0) = \sqrt{r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}}) + o(1) \Rightarrow \mathcal{N}(\mathbf{0}, \mathbf{I}_p), \quad a.s. \quad r \to \infty \quad and \quad \frac{r}{n} \to 0.$$

■

3.4 Massive Data Bootstrapping Confidence Interval

Interval estimate is as important as point estimate. These two types of statistical estimates when combined together can be considered as a good guess of the parameter. Efron(1993) concluded several ways of constructing bootstrap confidence intervals. The simplest one is the bootstrap quantile interval. A more robust one is the bootstrap-t confidence interval. We will modify the bootstrap-t confidence interval and construct the Massive Data Bootstrap confidence interval. In multivariate case, the confidence intervals become confidence region. In this section, we consider confidence intervals for each component of the multiple linear regression coefficient estimator. To not make the notations too complicated, without adding subscript, denote $\hat{\beta}$ as one of the components of $\hat{\boldsymbol{\beta}}$, and denote $\hat{\beta}^*$ as the corresponding component of $\hat{\boldsymbol{\beta}}^*$.

The bootstrap-t confidence interval for $\hat{\beta}$ is constructed as follows. Suppose we get B bootstrap samples from the original data,

$$(\mathbf{X}_1^{u*}, \mathbf{y}_1^{u*}), (\mathbf{X}_2^{u*}, \mathbf{y}_2^{u*}), ..., (\mathbf{X}_B^{u*}, \mathbf{y}_B^{u*}).$$

For each bootstrap sample, we can calculate two terms, $\hat{\beta}_b^{u*}$ and se_b^*, where the former is just the LSE from bootstrap sample $(\mathbf{X}_b^{u*}, \mathbf{y}_b^{u*})$, the latter is the bootstrap standard error of $\hat{\beta}_b^{u*}$, which can be obtained from the following second step bootstrap procedure. Take B_2 bootstrap samples from $(\mathbf{X}_b^{u*}, \mathbf{y}_b^{u*})$,

$$(\mathbf{X}_1^{u**}, \mathbf{y}_1^{u**}), (\mathbf{X}_2^{u**}, \mathbf{y}_2^{u**}), ..., (\mathbf{X}_{B_2}^{u**}, \mathbf{y}_{B_2}^{u**}),$$

calculate LSE $\hat{\beta}_{b_2}^{u**}$ for each bootstrap sample $(\mathbf{X}_{b_2}^{u**}, \mathbf{y}_{b_2}^{u**})$, $b_2 = 1, 2, ..., B_2$. Then se_b^* can be obtained by

$$se_b^* = \sqrt{\frac{1}{B_2 - 1} \sum_{b_2=1}^{B_2} (\hat{\beta}_{b_2}^{**} - \frac{1}{B_2} \sum_{b_2=1}^{B_2} \hat{\beta}_{b_2}^{**})^2}.$$

Once $\hat{\beta}_b^{u*}$ and se_b^* are calculated, one can get the bootstrap-t distribution which is an empirical distribution constructed by

$$F_B^*(x) = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}[t_b^* < x]$$

where

$$t_b^* = \frac{\hat{\beta}_b^{u*} - \hat{\beta}}{se_b^*}.$$

And t_α^* is such that

$$\#\{t^* : t_b^* \leq t_\alpha^*, b = 1, 2, ..., B\}/B = \alpha. \tag{3.4.1}$$

Now the bootstrap-t confidence interval for β_0 with confidence level $(1 - \alpha)$ can be constructed as

$$(\hat{\beta} - t_{1-\alpha/2}^* * \hat{se},\ \hat{\beta} - t_{\alpha/2}^* * \hat{se}), \tag{3.4.2}$$

where $\hat{se}$ is calculated as

$$\hat{se} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\beta}_b^* - \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^*)^2}.$$

As we can see from the bootstrap-t confidence interval, there are two stages of bootstrap, the first stage has repetition B and second stage has repetition B_2. This is time consuming, hence intuitively not the best way in massive data. Compared to traditional data, one advantage of massive data is that when we do subsampling there is still large enough sample size for the asymptotic results to hold. Therefore, we can apply the asymptotic results here to save time. Plus, in massive data case our focus is different from that of traditional sample size case. While our final task is still to estimate true parameter β_0, due to $\hat{\beta}$ being unknown, we are more interested in estimating $\hat{\beta}$ first. That is, we need to construct a $(1 - \alpha)$ confidence interval for $\hat{\beta}$. When the conditions of theorem 3.3.4 is satisfied, the confidence interval will also work for the true parameter value β_0. The $(1 - \alpha)$ confidence for $\hat{\beta}$ in massive data bootstrap will be constructed as follows. For massive data sample

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n),$$

draw a bootstrap sample with sampling probability $\boldsymbol{\pi}$,

$$(\mathbf{x}_1^*, y_1^*), (\mathbf{x}_2^*, y_2^*), ..., (\mathbf{x}_m^*, y_m^*)$$

with sample size $r << n$. Calculate WLSE

$$\hat{\boldsymbol{\beta}}^{*} = (\mathbf{X}^{*\top}\mathbf{W}^{*}\mathbf{X}^{*})^{-1}\mathbf{X}^{*\top}\mathbf{W}^{*}\mathbf{y}^{*}$$

and corresponding theoretical variance co-variance matrix

$$\mathbf{V} = \sqrt{(\mathbf{X}^{\top}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\top}\mathrm{Diag}(\frac{\hat{\boldsymbol{\varepsilon}}^{2}}{\boldsymbol{\pi}})\mathbf{X}(\mathbf{X}^{\top}\mathbf{X})^{-\mathbf{1}}}, \tag{3.4.3}$$

where $\mathbf{X}^{*} = (\mathbf{x}_1^{*}, \mathbf{x}_2^{*}, ..., \mathbf{x}_m^{*})^{\top}$, $\mathbf{y}^{*} = (y_1^{*}, y_2^{*}, ..., y_m^{*})^{\top}$, $\hat{\boldsymbol{\varepsilon}}$ is the residual vector. Then we can calculate $\hat{se}$ by taking the square root of diagonal elements of the $\mathbf{V}$ for the corresponding component of $\hat{\boldsymbol{\beta}}^{*}$ and the confidence interval for the component $\hat{\beta}^{*}$ is constructed as

$$(\hat{\beta}^{*} - z_{1-\alpha} * \hat{se},\ \hat{\beta}^{*} - z_{\alpha} * \hat{se}), \tag{3.4.4}$$

When $\boldsymbol{\pi} = (\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n})^{\top}$, (3.4.4) becomes

$$(\hat{\beta}^{u*} - z_{1-\alpha} * \hat{se}^{u},\ \hat{\beta}^{u*} - z_{\alpha} * \hat{se}^{u}), \tag{3.4.5}$$

when $\boldsymbol{\pi}$ is the optimal sampling probability, (3.4.4) becomes

$$(\hat{\beta}^{o*} - z_{1-\alpha} * \hat{se}^{o},\ \hat{\beta}^{o*} - z_{\alpha} * \hat{se}^{o}). \tag{3.4.6}$$

4. SIMULATION STUDY

In this chapter, we perform numerical study of k-means clustering and bootstrapping via optimal subsampling algorithms.

4.1 Simulation Study for Massive Data K-means Clustering

In this section, we compare the k-means clustering via uniform sampling and optimal subsampling by presenting the MSE of both centroid vectors. Time ratio is also presented in the tables and compared. The comparison is divided into two sub-sections: equal cluster size case and unequal cluster size case.

4.1.1 Equal Cluster Size Case

In this case, data are simulated from isotropic Gaussian blobs using the make_bulbs function in Python3.7. Data are generated from different combinations of k and d. k =3, 6, 9 and 12. d = 5, 15 and 25. Full sample size $n = 1,000,000$. Subsample size r and presample size r_0 vary in the following combinations: $(r, r_0) = (0.2n, 0.05n), (0.1n, 0.05n), (0.05n, 0.05n)$, $(0.05n, 0.01n)$ and $(0.01n, 0.01n)$. Cluster standard deviation σ is a hyper-parameter for generating the Gaussian blobs. The data with smaller σ will have more separated blobs while the data with larger σ will have blobs that cover each others' area and are hard to be correctly clustered. In our simulation study we choose three cases: σ=0.5, 1 and 1.5. The MSE's of the centroid vectors $\mathbf{b}^*$ from uniform and A-optimal subsampling are compared, where

$$MSE(\mathbf{b}^*) = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{b}^* - \hat{\mathbf{b}}||^2,$$

$M = 100$ is the number of repetitions. Time ratio of the subsample calculation to full sample calculation is also compared between the uniform and A-optimal subsampling.

Table 4.1.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 0.5, d = 5, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	5	0.20	0.05	0.00005	0.00005	0.25641	0.49353
3	5	0.10	0.05	0.00011	0.00010	0.17685	0.41251
3	5	0.05	0.05	0.00023	0.00021	0.13964	0.37009
3	5	0.05	0.01	0.00024	0.00021	0.14133	0.34107
3	5	0.01	0.01	0.00113	0.00106	0.10856	0.29938
6	5	0.20	0.05	0.00022	0.00019	0.22602	0.39526
6	5	0.10	0.05	0.00044	0.00040	0.15848	0.33574
6	5	0.05	0.05	0.00092	0.00083	0.11685	0.28814
6	5	0.05	0.01	0.00084	0.00085	0.11441	0.25604
6	5	0.01	0.01	0.00446	0.00424	0.08808	0.23017
9	5	0.20	0.05	0.00051	0.00047	0.21124	0.34148
9	5	0.10	0.05	0.00102	0.00093	0.12270	0.24427
9	5	0.05	0.05	0.00200	0.00191	0.08908	0.21269
9	5	0.05	0.01	0.00205	0.00190	0.08834	0.18269
9	5	0.01	0.01	0.00992	0.00912	0.06075	0.15467
12	5	0.20	0.05	0.00088	0.00080	0.20503	0.32069
12	5	0.10	0.05	0.00175	0.00163	0.12245	0.23475
12	5	0.05	0.05	0.00355	0.00331	0.08240	0.19222
12	5	0.05	0.01	0.00373	0.00337	0.08285	0.16326
12	5	0.01	0.01	0.01765	0.01628	0.05465	0.13486

Table 4.2.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1, d = 5, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	5	0.20	0.05	0.00023	0.00019	0.25942	0.49726
3	5	0.10	0.05	0.00046	0.00040	0.17296	0.39689
3	5	0.05	0.05	0.00088	0.00075	0.13571	0.35885
3	5	0.05	0.01	0.00089	0.00079	0.13469	0.32504
3	5	0.01	0.01	0.00455	0.00409	0.10481	0.28930
6	5	0.20	0.05	0.00093	0.00081	0.23098	0.39426
6	5	0.10	0.05	0.00189	0.00172	0.14770	0.30858
6	5	0.05	0.05	0.00348	0.00317	0.10490	0.25794
6	5	0.05	0.01	0.00366	0.00317	0.09548	0.20451
6	5	0.01	0.01	0.01823	0.01586	0.07147	0.18571
9	5	0.20	0.05	0.00206	0.00187	0.21489	0.34099
9	5	0.10	0.05	0.00407	0.00350	0.12495	0.24472
9	5	0.05	0.05	0.00795	0.00756	0.08487	0.20057
9	5	0.05	0.01	0.00823	0.00749	0.08512	0.17156
9	5	0.01	0.01	0.03880	0.03587	0.05526	0.13809
12	5	0.20	0.05	0.00363	0.00324	0.20750	0.31979
12	5	0.10	0.05	0.00724	0.00629	0.11999	0.22693
12	5	0.05	0.05	0.01410	0.01280	0.07945	0.18324
12	5	0.05	0.01	0.01440	0.01314	0.08010	0.15357
12	5	0.01	0.01	0.07206	0.06560	0.04852	0.11975

Table 4.3.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1.5, d = 5, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	5	0.20	0.05	0.00050	0.00045	0.23199	0.42863
3	5	0.10	0.05	0.00104	0.00099	0.15493	0.34741
3	5	0.05	0.05	0.00193	0.00187	0.12020	0.30875
3	5	0.05	0.01	0.00208	0.00182	0.11931	0.27706
3	5	0.01	0.01	0.01122	0.00918	0.08802	0.24352
6	5	0.20	0.05	0.00197	0.00186	0.24717	0.39988
6	5	0.10	0.05	0.00389	0.00373	0.14464	0.28645
6	5	0.05	0.05	0.00811	0.00729	0.10148	0.24870
6	5	0.05	0.01	0.00821	0.00727	0.10059	0.20692
6	5	0.01	0.01	0.04078	0.03627	0.06435	0.16638
9	5	0.20	0.05	0.00450	0.00416	0.22010	0.33510
9	5	0.10	0.05	0.00945	0.00851	0.12684	0.23720
9	5	0.05	0.05	0.01816	0.01610	0.08059	0.18390
9	5	0.05	0.01	0.01876	0.01664	0.08328	0.15937
9	5	0.01	0.01	0.08771	0.08146	0.04878	0.12149
12	5	0.20	0.05	0.00821	0.00738	0.20443	0.31040
12	5	0.10	0.05	0.01626	0.01437	0.11698	0.21318
12	5	0.05	0.05	0.03279	0.02896	0.07172	0.16753
12	5	0.05	0.01	0.03240	0.02873	0.07285	0.13335
12	5	0.01	0.01	0.15746	0.14743	0.03995	0.09705

Table 4.4.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 0.5, d = 15, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	15	0.20	0.05	0.00017	0.00016	0.24267	0.44565
3	15	0.10	0.05	0.00033	0.00032	0.15561	0.35308
3	15	0.05	0.05	0.00064	0.00066	0.11231	0.30970
3	15	0.05	0.01	0.00068	0.00067	0.10809	0.26705
3	15	0.01	0.01	0.00346	0.00316	0.07967	0.23895
6	15	0.20	0.05	0.00070	0.00067	0.23225	0.39124
6	15	0.10	0.05	0.00136	0.00130	0.13953	0.29295
6	15	0.05	0.05	0.00272	0.00267	0.09391	0.24520
6	15	0.05	0.01	0.00273	0.00264	0.09437	0.21199
6	15	0.01	0.01	0.01344	0.01312	0.05839	0.16653
9	15	0.20	0.05	0.00152	0.00142	0.21325	0.33815
9	15	0.10	0.05	0.00310	0.00292	0.12295	0.24360
9	15	0.05	0.05	0.00617	0.00583	0.08015	0.19754
9	15	0.05	0.01	0.00611	0.00574	0.07966	0.16700
9	15	0.01	0.01	0.03037	0.02927	0.04932	0.13383
12	15	0.20	0.05	0.00270	0.00265	0.20862	0.32126
12	15	0.10	0.05	0.00540	0.00521	0.11992	0.22704
12	15	0.05	0.05	0.01051	0.01037	0.07547	0.18078
12	15	0.05	0.01	0.01070	0.01050	0.07593	0.15151
12	15	0.01	0.01	0.05409	0.05181	0.04468	0.11748

Table 4.5.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1, d = 15, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	15	0.20	0.05	0.00068	0.00065	0.23936	0.44026
3	15	0.10	0.05	0.00138	0.00132	0.15409	0.34984
3	15	0.05	0.05	0.00266	0.00257	0.10733	0.29452
3	15	0.05	0.01	0.00283	0.00268	0.10849	0.26996
3	15	0.01	0.01	0.01309	0.01353	0.08090	0.24089
6	15	0.20	0.05	0.00272	0.00256	0.22930	0.38676
6	15	0.10	0.05	0.00560	0.00531	0.13774	0.28833
6	15	0.05	0.05	0.01079	0.01054	0.09615	0.24792
6	15	0.05	0.01	0.01051	0.01045	0.09351	0.21247
6	15	0.01	0.01	0.05350	0.05125	0.05998	0.17172
9	15	0.20	0.05	0.00604	0.00580	0.20876	0.33073
9	15	0.10	0.05	0.01208	0.01140	0.12815	0.25279
9	15	0.05	0.05	0.02402	0.02335	0.07952	0.19695
9	15	0.05	0.01	0.02431	0.02369	0.07886	0.16514
9	15	0.01	0.01	0.12025	0.11922	0.04880	0.13285
12	15	0.20	0.05	0.01070	0.01028	0.21273	0.32400
12	15	0.10	0.05	0.02114	0.02059	0.12257	0.23330
12	15	0.05	0.05	0.04315	0.04212	0.07629	0.18359
12	15	0.05	0.01	0.04267	0.04229	0.07801	0.15416
12	15	0.01	0.01	0.21314	0.20848	0.04658	0.12200

Table 4.6.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1.5, d = 15, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	15	0.20	0.05	0.00153	0.00145	0.23419	0.42452
3	15	0.10	0.05	0.00300	0.00286	0.13984	0.31451
3	15	0.05	0.05	0.00599	0.00613	0.10362	0.28566
3	15	0.05	0.01	0.00614	0.00596	0.10452	0.25502
3	15	0.01	0.01	0.03086	0.02905	0.07550	0.22583
6	15	0.20	0.05	0.00619	0.00595	0.23614	0.39261
6	15	0.10	0.05	0.01190	0.01137	0.14533	0.30129
6	15	0.05	0.05	0.02391	0.02355	0.09254	0.23771
6	15	0.05	0.01	0.02437	0.02328	0.09336	0.20834
6	15	0.01	0.01	0.12379	0.11791	0.06234	0.17748
9	15	0.20	0.05	0.01352	0.01311	0.22762	0.35315
9	15	0.10	0.05	0.02738	0.02592	0.13212	0.25861
9	15	0.05	0.05	0.05399	0.05233	0.08191	0.19979
9	15	0.05	0.01	0.05496	0.05258	0.08229	0.16782
9	15	0.01	0.01	0.27130	0.26505	0.04720	0.12794
12	15	0.20	0.05	0.02395	0.02356	0.23742	0.35552
12	15	0.10	0.05	0.04833	0.04692	0.13116	0.24607
12	15	0.05	0.05	0.09677	0.09300	0.08204	0.19513
12	15	0.05	0.01	0.09563	0.09463	0.08025	0.15286
12	15	0.01	0.01	0.47289	0.45823	0.04527	0.11815

Table 4.7.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 0.5, d = 25, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	25	0.20	0.05	0.00028	0.00027	0.27032	0.45907
3	25	0.10	0.05	0.00056	0.00055	0.14559	0.32627
3	25	0.05	0.05	0.00113	0.00110	0.09778	0.27439
3	25	0.05	0.01	0.00112	0.00113	0.09732	0.24062
3	25	0.01	0.01	0.00570	0.00544	0.06556	0.20688
6	25	0.20	0.05	0.00113	0.00110	0.24696	0.39672
6	25	0.10	0.05	0.00222	0.00218	0.13215	0.27412
6	25	0.05	0.05	0.00448	0.00435	0.08473	0.22368
6	25	0.05	0.01	0.00448	0.00446	0.08656	0.19555
6	25	0.01	0.01	0.02274	0.02228	0.05350	0.16034
9	25	0.20	0.05	0.00253	0.00249	0.23328	0.35620
9	25	0.10	0.05	0.00501	0.00492	0.12077	0.23471
9	25	0.05	0.05	0.01010	0.00983	0.07465	0.18892
9	25	0.05	0.01	0.00995	0.00996	0.07640	0.16062
9	25	0.01	0.01	0.05149	0.04840	0.04411	0.12652
12	25	0.20	0.05	0.00449	0.00443	0.21276	0.31801
12	25	0.10	0.05	0.00906	0.00885	0.11880	0.22412
12	25	0.05	0.05	0.01790	0.01761	0.07193	0.17505
12	25	0.05	0.01	0.01796	0.01764	0.07147	0.14195
12	25	0.01	0.01	0.08969	0.08876	0.04057	0.11205

Table 4.8.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1, d = 25, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	25	0.20	0.05	0.00113	0.00110	0.25190	0.42810
3	25	0.10	0.05	0.00226	0.00221	0.14794	0.33192
3	25	0.05	0.05	0.00453	0.00442	0.09560	0.26957
3	25	0.05	0.01	0.00456	0.00443	0.09938	0.24529
3	25	0.01	0.01	0.02254	0.02144	0.06401	0.20214
6	25	0.20	0.05	0.00456	0.00441	0.23025	0.36803
6	25	0.10	0.05	0.00909	0.00883	0.12156	0.25337
6	25	0.05	0.05	0.01802	0.01743	0.08131	0.21435
6	25	0.05	0.01	0.01801	0.01786	0.08002	0.18127
6	25	0.01	0.01	0.08885	0.08622	0.05020	0.14973
9	25	0.20	0.05	0.00997	0.00981	0.23207	0.35312
9	25	0.10	0.05	0.02027	0.01957	0.12131	0.23860
9	25	0.05	0.05	0.04007	0.03972	0.07802	0.19605
9	25	0.05	0.01	0.04036	0.03996	0.07908	0.16444
9	25	0.01	0.01	0.20379	0.19925	0.04341	0.12487
12	25	0.20	0.05	0.01792	0.01760	0.21798	0.32774
12	25	0.10	0.05	0.03581	0.03525	0.11659	0.21936
12	25	0.05	0.05	0.07183	0.06996	0.07383	0.17904
12	25	0.05	0.01	0.07175	0.07094	0.07346	0.14464
12	25	0.01	0.01	0.35684	0.35129	0.03948	0.10918

Table 4.9.: Massive Data k-means Clustering Comparison in Equal Cluster Size, $\sigma = 1.5, d = 25, n = 1,000,000$

k	d	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		TimeRatio	
				Unif	Opt	Unif	Opt
3	25	0.20	0.05	0.00255	0.00237	0.26990	0.45634
3	25	0.10	0.05	0.00522	0.00493	0.14820	0.33126
3	25	0.05	0.05	0.01017	0.00993	0.09468	0.26511
3	25	0.05	0.01	0.01009	0.00990	0.09781	0.24004
3	25	0.01	0.01	0.05162	0.05014	0.06338	0.20078
6	25	0.20	0.05	0.01010	0.00988	0.25985	0.40296
6	25	0.10	0.05	0.02007	0.01957	0.13630	0.27932
6	25	0.05	0.05	0.04019	0.03901	0.08768	0.22898
6	25	0.05	0.01	0.04029	0.03884	0.08858	0.19373
6	25	0.01	0.01	0.20096	0.19758	0.05197	0.15622
9	25	0.20	0.05	0.02249	0.02244	0.23920	0.36171
9	25	0.10	0.05	0.04571	0.04497	0.13096	0.24956
9	25	0.05	0.05	0.09091	0.08998	0.07992	0.19630
9	25	0.05	0.01	0.09065	0.09000	0.07891	0.16019
9	25	0.01	0.01	0.44923	0.44654	0.04336	0.12251
12	25	0.20	0.05	0.04071	0.03949	0.22706	0.33770
12	25	0.10	0.05	0.08123	0.07890	0.12272	0.22351
12	25	0.05	0.05	0.16084	0.15739	0.07362	0.17516
12	25	0.05	0.01	0.16235	0.15683	0.07039	0.13533
12	25	0.01	0.01	0.80132	0.78467	0.03627	0.09866

From the output table (4.1) to table (4.9), we can see that, for different k and combinations of r and r_0, the MSE of the centroid estimator from A-optimal subsampling is generally smaller than that of uniform subsampling. When comparing the time ratios, we can see the computation times of the A-optimal subsampling are longer but acceptable. In conclusion, the A-optimal subsampling outperforms the uniform subsampling in the k-means analysis with smaller MSE, while the computation times are comparable.

4.1.2 Unequal Cluster Size Case

This is a more realistic case, for example, the clusters of different news topics may contain different number of words. In this case, data are also simulated from isotropic Gaussian blobs using the make_bulbs function in Python3.7. Three different data are generated. For purpose of better data visualization, we choose dimension $d = 2$ for the three simulated data sets. Number of clusters k vary in 3, 4 and 5. Since the plot of the data could be too massy if number of observations n is too large, we choose the value $n = 100,000$ and $1,000,000$. Subsample size r and presample size r_0 vary in the following combinations: $(r, r_0) = (0.01n, 0.005n), (0.05n, 0.005n), (0.1n, 0.05n)$ and $(0.2n, 0.05n)$. The MSE's of the centroid vectors $\mathbf{b}^*$ from uniform and A-optimal sampling are compared, where

$$MSE(\mathbf{b}^*) = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{b}^* - \hat{\mathbf{b}}||^2,$$

$M = 100$ is the number of repetitions. Time ratios of the subsample calculation to full sample calculation are also compared between the uniform and A-optimal subsampling.

Table 4.10.: Massive Data k-means Clustering Comparison in Unequal Cluster Size, $k = 3, d = 2$

n	r	r_0	MSE		TimeRatio	
			Unif	Opt	Unif	Opt
100,000	1000	500	1.07833	0.80578	0.05269	0.10144
100,000	5000	500	0.90437	0.70793	0.08643	0.12825
100,000	10,000	5000	0.99572	0.62584	0.11707	0.18923
100,000	20,000	5000	0.87321	0.53229	0.20531	0.25758
1,000,000	10,000	5000	0.70149	0.26682	0.02878	0.07630
1,000,000	50,000	5000	0.41199	0.00012	0.06830	0.10084
1,000,000	100,000	50,000	0.14297	0.00005	0.12227	0.17950
1,000,000	200,000	50,000	0.26435	0.00004	0.21704	0.23332

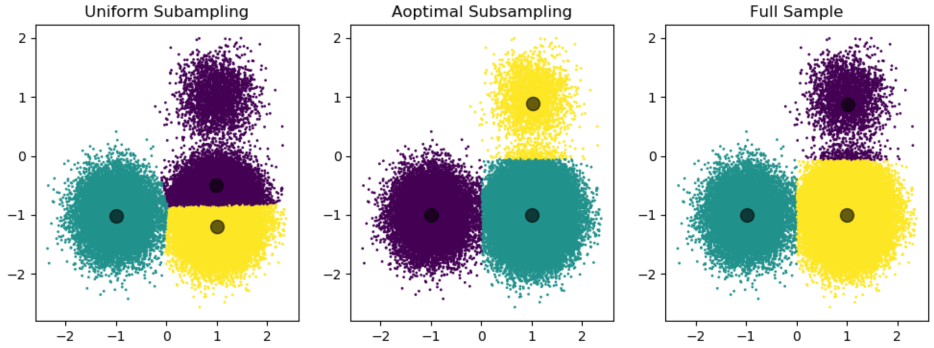
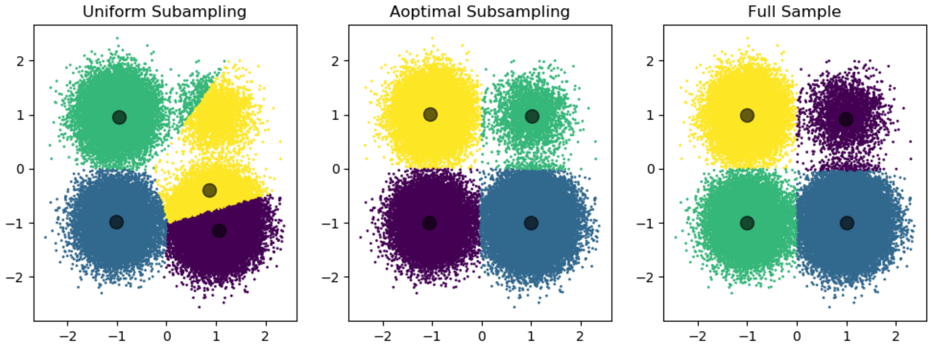


Figure 4.1.: Massive Data k-means Clustering Visualization, $k = 3, d = 2$

Table 4.11.: Massive Data k-means Clustering Comparison in Unequal Cluster Size, $k = 4, d = 2$

n	r	r_0	MSE		TimeRatio	
			Unif	Opt	Unif	Opt
100,000	1000	500	0.15489	0.10082	0.03000	0.05913
100,000	5000	500	0.03380	0.02936	0.05902	0.08457
100,000	10,000	5000	0.01704	0.00703	0.09185	0.14273
100,000	20,000	5000	0.00933	0.00457	0.21657	0.24049
1,000,000	10,000	5000	0.00129	0.00049	0.04286	0.11363
1,000,000	50,000	5000	0.00032	0.00011	0.09710	0.15758
1,000,000	100,000	50,000	0.00015	0.00006	0.16490	0.26183
1,000,000	200,000	50,000	0.00009	0.00003	0.30577	0.37170


Uniform Subampling

Aoptimal Subsampling

Full Sample

Figure 4.2.: Massive Data k-means Clustering Visualization, $k = 4, d = 2$

Table 4.12.: Massive Data k-means Clustering Comparison in Unequal Cluster Size, $k = 5, d = 2$

n	r	r_0	MSE		TimeRatio	
			Unif	Opt	Unif	Opt
100,000	1000	500	0.40595	0.07203	0.04740	0.09316
100,000	5000	500	0.14333	0.00122	0.08088	0.11970
100,000	10,000	5000	0.13703	0.00039	0.12454	0.18693
100,000	20,000	5000	0.24277	0.00018	0.26671	0.27604
1,000,000	10,000	5000	0.10143	0.00035	0.02288	0.05815
1,000,000	50,000	5000	0.08218	0.00006	0.05914	0.08066
1,000,000	100,000	50,000	0.10614	0.00003	0.11548	0.15951
1,000,000	200,000	50,000	0.12995	0.00002	0.21637	0.22689

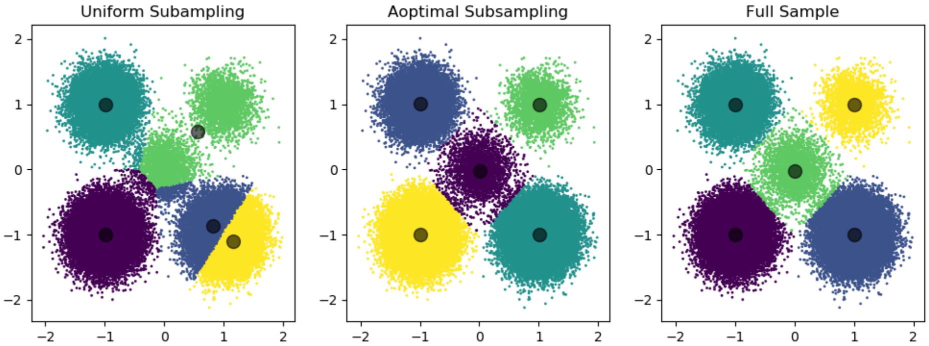


Figure 4.3.: Massive Data k-means Clustering Visualization, $k = 5, d = 2$

From table (4.10), table (4.11) and table (4.12) we can see that that in the more realistic situation of unequal cluster size, the MSE's of the centroid vector from A-optimal subsampling are always smaller than that of uniform subsampling. The difference becomes even larger when number of clusters increases from 3 to 5. Visualized from figure (4.1), figure (4.2) and figure (4.3), the result is more clear: the k-means clusters from uniform subsampling deviate from full sample k-means clustering result while A-optimal subsampling gives consistent result with full sample.

4.2 Simulation Study for Massive Data Bootstrapping

In this section, we perform the simulation study for massive data bootstrapping via A-optimal subsampling. The coverage probabilities and lengths of confidence intervals, the standard errors and running times are compared.

4.2.1 Confidence Interval Comparison

We focus on confidence interval constructions and compare lengths and coverage probabilities of the confidence intervals while controlling the confidence level. To be specific, set the nominal confidence level to 95%, we first compare the coverage probabilities of both regular uniform bootstrapping method and the proposed A-optimal bootstrapping method to the nominal confidence level. Only when the coverage probabilities between the two methods are comparable and close to nominal, it makes sense to further compare them in length of confidence intervals. If the coverage probabilities of the confidence intervals from both methods are close to the nominal confidence level, the shorter confidence interval will indicate a more efficient estimator. From another perspective, the more efficient the estimator, the smaller the required sample size for constructing confidence intervals of the same length.

Observations $\mathbf{x}_i$, $i = 1, 2, ..., n$ are generated from three different p-dimension multivariate distributions: GA(multivariate Gaussian distribution), LN(multivariate Log-normal distribution) and T3(multivariate t distribution with 3 degrees of freedom). Error terms ε_i, $i = 1, 2, ..., n$ are generated from four different distributions: GA(Gaussian distribution), LN (Log-normal distribution), T3 (student t distribution with 3 degrees of freedom) and LAP(Laplace distribution). All distributions mentioned above have the origin location parameter and unit valued scale parameter. Sample size n vary in 100,000, 500,000 to 1,000,000. Dimension p vary in 10, 30 and

50. The massive data bootstrap sample size m is from 1000, 5000 to 10,000. $\boldsymbol{\beta}_0$ is a $(p+1)$ dimension vector, of which the first to $([\frac{p}{2}]+1)$th components are 1, the rest of the components are -1. The confidence intervals are constructed using Algorithm 2 and 3, and formula (3.4.5) and (3.4.6), where the massive data bootstrap uniform subsampling uses equal sampling probabilities and A-optimal subsampling uses sampling probabilities from Theorem 3.3.3. And the tables are based on the second components of $\boldsymbol{\beta}_0$ and $\hat{\boldsymbol{\beta}}$, denoted as β_0 and $\hat{\beta}$ (we keep the notation consistent with Chapter 3).

Due to the large number of combinations from the above parameter values, we divide the results into 12 tables, i.e., table (4.13) to table (4.24). In each table, the meaning of the columns are:

- n: sample size of full data. Values: 100k, 500k, 1M;
- p: number of features of full data. Values: 10, 30, 50;
- m: massive data bootstrap sample size. Values: 1000, 5000, 10,000;
- CP $\hat{\beta}$: coverage probabilities about $\hat{\beta}$;
- CP β_0:coverage probabilities about β_0;
- len: length of confidence intervals;
- $\hat{se}$.th: theoretical standard error of $\hat{\beta}^*$ from taking square root of the second diagonal element of formula (3.4.3);
- $\hat{se}$.data: empirical standard error of $\hat{\beta}^*$ from massive data bootstrap samples.

In each table, there are 27 rows, each row is a different scenario according to the value of n, p and m. For every single scenario we compare massive data bootstrap via uniform subsampling and that via A-optimal subsampling.

From table (4.13) to table (4.16), we can see that the coverage probabilities for β_0 and $\hat{\beta}$ are always close to 95% for massive data bootstrap with uniform subsampling.

Under the A-optimal subsampling case, most of the coverage probabilities are close to 95% except for when m is too small compared to n (for example, $n = 1,000,000, m = 1000$, i.e. $m = 0.001 * n$) or when the number of features p is very large (50). In both cases the coverage probabilities for both β_0 and $\hat{\beta}$ could be slightly smaller than the nominal level. There is a reason for this to happen: the A-optimal sampling probability is obtained by minimizing the leading term of the asymptotic variance, and our theory is based on fixed p. So in the case of large p, the higher order term can not be ignored, and the variance could be underestimated. Our suggestion from this simulation study is that, in order to get a good interval estimate of β_0 and $\hat{\beta}$ the massive data bootstrap sample size m should be greater than 1% of the full sample when the full sample size n is around one million, especially if p is large (for example, 50).

Table 4.13.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim GA, \varepsilon \sim GA$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9486	0.9391	0.9477	0.9369	0.2073	0.1615	0.0529	0.0412	0.0531	0.0431
100,000	10	5000	0.9514	0.9479	0.9460	0.9369	0.0927	0.0725	0.0236	0.0185	0.0235	0.0187
100,000	10	10,000	0.9494	0.9473	0.9386	0.9316	0.0654	0.0512	0.0167	0.0131	0.0167	0.0132
100,000	30	1000	0.9461	0.9276	0.9449	0.9265	0.2076	0.1646	0.0530	0.0420	0.0538	0.0457
100,000	30	5000	0.9498	0.9451	0.9423	0.9319	0.0928	0.0737	0.0237	0.0188	0.0237	0.0191
100,000	30	10,000	0.9521	0.9484	0.9384	0.9290	0.0658	0.0521	0.0168	0.0133	0.0166	0.0134
100,000	50	1000	0.9460	0.9197	0.9449	0.9181	0.2083	0.1655	0.0531	0.0422	0.0539	0.0472
100,000	50	5000	0.9487	0.9441	0.9406	0.9325	0.0929	0.0741	0.0237	0.0189	0.0238	0.0194
100,000	50	10,000	0.9489	0.9448	0.9370	0.9255	0.0658	0.0522	0.0168	0.0133	0.0168	0.0136
500,000	10	1000	0.9499	0.9396	0.9503	0.9395	0.2068	0.1620	0.0528	0.0413	0.0529	0.0430
500,000	10	5000	0.9474	0.9488	0.9472	0.9472	0.0924	0.0725	0.0236	0.0185	0.0238	0.0187
500,000	10	10,000	0.9493	0.9473	0.9477	0.9441	0.0654	0.0512	0.0167	0.0131	0.0167	0.0131
500,000	30	1000	0.9454	0.9272	0.9457	0.9269	0.2065	0.1640	0.0527	0.0418	0.0535	0.0456
500,000	30	5000	0.9505	0.9432	0.9498	0.9411	0.0926	0.0734	0.0236	0.0187	0.0236	0.0192
500,000	30	10,000	0.9494	0.9471	0.9471	0.9420	0.0655	0.0519	0.0167	0.0132	0.0167	0.0134
500,000	50	1000	0.9442	0.9154	0.9440	0.9150	0.2068	0.1646	0.0528	0.0420	0.0539	0.0475
500,000	50	5000	0.9493	0.9403	0.9475	0.9390	0.0924	0.0736	0.0236	0.0188	0.0237	0.0194
500,000	50	10,000	0.9492	0.9452	0.9455	0.9395	0.0655	0.0520	0.0167	0.0133	0.0167	0.0135
1,000,000	10	1000	0.9473	0.9402	0.9469	0.9402	0.2066	0.1618	0.0527	0.0413	0.0532	0.0429
1,000,000	10	5000	0.9496	0.9470	0.9496	0.9459	0.0924	0.0723	0.0236	0.0185	0.0236	0.0187
1,000,000	10	10,000	0.9502	0.9484	0.9493	0.9471	0.0654	0.0511	0.0167	0.0130	0.0167	0.0131
1,000,000	30	1000	0.9453	0.9268	0.9452	0.9263	0.2065	0.1641	0.0527	0.0419	0.0538	0.0458
1,000,000	30	5000	0.9506	0.9447	0.9495	0.9433	0.0925	0.0734	0.0236	0.0187	0.0237	0.0192
1,000,000	30	10,000	0.9500	0.9478	0.9487	0.9469	0.0654	0.0519	0.0167	0.0132	0.0167	0.0133
1,000,000	50	1000	0.9419	0.9169	0.9420	0.9161	0.2068	0.1644	0.0528	0.0419	0.0543	0.0474
1,000,000	50	5000	0.9473	0.9410	0.9467	0.9410	0.0925	0.0736	0.0236	0.0188	0.0237	0.0194
1,000,000	50	10,000	0.9502	0.9460	0.9488	0.9435	0.0654	0.0520	0.0167	0.0133	0.0167	0.0135

From table (4.13) we can see with coverage probabilities close to the nominal confidence level, the length of the A-optimal subsampling confidence interval of the massive data bootstrap regression estimator is shorter than that of uniform subsampling. The standard error of the former is also smaller. Both methods' empirical standard errors are close to theoretical standard errors when $\frac{m}{p}$ is larger than 100.

Table 4.14.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim GA, \varepsilon \sim LN$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9488	0.9405	0.9472	0.9361	0.4475	0.2556	0.1142	0.0652	0.1146	0.0679
100,000	10	5000	0.9495	0.9475	0.9453	0.9320	0.2000	0.1146	0.0510	0.0292	0.0507	0.0296
100,000	10	10,000	0.9487	0.9486	0.9365	0.9125	0.1409	0.0811	0.0360	0.0207	0.0360	0.0208
100,000	30	1000	0.9450	0.9277	0.9442	0.9237	0.4465	0.2601	0.1139	0.0664	0.1156	0.0723
100,000	30	5000	0.9500	0.9455	0.9432	0.9221	0.1997	0.1166	0.0510	0.0298	0.0510	0.0304
100,000	30	10,000	0.9523	0.9477	0.9390	0.9132	0.1423	0.0825	0.0363	0.0210	0.0359	0.0212
100,000	50	1000	0.9437	0.9176	0.9426	0.9133	0.4499	0.2616	0.1148	0.0667	0.1174	0.0753
100,000	50	5000	0.9495	0.9445	0.9445	0.9299	0.2019	0.1174	0.0515	0.0300	0.0515	0.0307
100,000	50	10,000	0.9488	0.9443	0.9369	0.9090	0.1424	0.0827	0.0363	0.0211	0.0363	0.0215
500,000	10	1000	0.9487	0.9409	0.9483	0.9403	0.4471	0.2565	0.1141	0.0654	0.1144	0.0680
500,000	10	5000	0.9477	0.9479	0.9469	0.9443	0.1992	0.1147	0.0508	0.0293	0.0511	0.0294
500,000	10	10,000	0.9506	0.9498	0.9490	0.9439	0.1411	0.0809	0.0360	0.0207	0.0359	0.0206
500,000	30	1000	0.9471	0.9272	0.9471	0.9266	0.4468	0.2597	0.1140	0.0663	0.1155	0.0722
500,000	30	5000	0.9505	0.9437	0.9491	0.9392	0.2001	0.1161	0.0511	0.0296	0.0510	0.0303
500,000	30	10,000	0.9512	0.9455	0.9483	0.9392	0.1413	0.0821	0.0361	0.0210	0.0360	0.0212
500,000	50	1000	0.9439	0.9165	0.9437	0.9159	0.4476	0.2604	0.1142	0.0664	0.1170	0.0753
500,000	50	5000	0.9494	0.9418	0.9481	0.9396	0.1995	0.1165	0.0509	0.0297	0.0512	0.0307
500,000	50	10,000	0.9494	0.9458	0.9466	0.9370	0.1416	0.0823	0.0361	0.0210	0.0361	0.0214
1,000,000	10	1000	0.9476	0.9394	0.9473	0.9391	0.4475	0.2560	0.1142	0.0653	0.1149	0.0680
1,000,000	10	5000	0.9492	0.9463	0.9490	0.9449	0.1994	0.1145	0.0509	0.0292	0.0509	0.0295
1,000,000	10	10,000	0.9493	0.9490	0.9489	0.9452	0.1415	0.0809	0.0361	0.0207	0.0360	0.0208
1,000,000	30	1000	0.9446	0.9271	0.9446	0.9270	0.4463	0.2597	0.1139	0.0662	0.1157	0.0725
1,000,000	30	5000	0.9488	0.9444	0.9484	0.9419	0.1997	0.1161	0.0509	0.0296	0.0510	0.0303
1,000,000	30	10,000	0.9496	0.9471	0.9489	0.9454	0.1416	0.0821	0.0361	0.0210	0.0361	0.0212
1,000,000	50	1000	0.9435	0.9127	0.9433	0.9124	0.4477	0.2600	0.1142	0.0663	0.1178	0.0759
1,000,000	50	5000	0.9475	0.9421	0.9472	0.9402	0.2001	0.1166	0.0511	0.0297	0.0513	0.0308
1,000,000	50	10,000	0.9489	0.9442	0.9474	0.9415	0.1412	0.0823	0.0360	0.0210	0.0363	0.0215

The results and conclusions of table (4.14) are similar to those of table (4.13).

Table 4.15.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim GA, \varepsilon \sim T3$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9490	0.9397	0.9484	0.9356	0.3557	0.2239	0.0907	0.0571	0.0914	0.0596
100,000	10	5000	0.9515	0.9470	0.9464	0.9328	0.1601	0.1001	0.0409	0.0255	0.0407	0.0259
100,000	10	10,000	0.9506	0.9504	0.9391	0.9237	0.1129	0.0709	0.0288	0.0181	0.0288	0.0181
100,000	30	1000	0.9458	0.9216	0.9452	0.9186	0.3584	0.2279	0.0914	0.0581	0.0932	0.0646
100,000	30	5000	0.9482	0.9439	0.9449	0.9341	0.1611	0.1016	0.0411	0.0259	0.0413	0.0266
100,000	30	10,000	0.9512	0.9480	0.9345	0.9079	0.1119	0.0720	0.0286	0.0184	0.0285	0.0185
100,000	50	1000	0.9455	0.9091	0.9450	0.9067	0.3601	0.2288	0.0919	0.0584	0.0935	0.0676
100,000	50	5000	0.9501	0.9392	0.9463	0.9264	0.1599	0.1022	0.0408	0.0261	0.0407	0.0272
100,000	50	10,000	0.9512	0.9465	0.9406	0.9174	0.1133	0.0724	0.0289	0.0185	0.0288	0.0188
500,000	10	1000	0.9484	0.9363	0.9480	0.9361	0.3556	0.2235	0.0907	0.0570	0.0915	0.0603
500,000	10	5000	0.9489	0.9467	0.9476	0.9430	0.1603	0.1000	0.0409	0.0255	0.0410	0.0259
500,000	10	10,000	0.9499	0.9474	0.9457	0.9387	0.1131	0.0707	0.0289	0.0180	0.0288	0.0182
500,000	30	1000	0.9466	0.9216	0.9465	0.9212	0.3576	0.2271	0.0912	0.0579	0.0923	0.0644
500,000	30	5000	0.9504	0.9412	0.9485	0.9391	0.1594	0.1015	0.0407	0.0259	0.0407	0.0267
500,000	30	10,000	0.9477	0.9456	0.9451	0.9378	0.1132	0.0718	0.0289	0.0183	0.0291	0.0187
500,000	50	1000	0.9441	0.9089	0.9439	0.9079	0.3598	0.2275	0.0918	0.0580	0.0947	0.0674
500,000	50	5000	0.9494	0.9390	0.9485	0.9366	0.1623	0.1018	0.0414	0.0260	0.0417	0.0271
500,000	50	10,000	0.9498	0.9452	0.9474	0.9390	0.1132	0.0720	0.0289	0.0184	0.0288	0.0189
1,000,000	10	1000	0.9491	0.9371	0.9488	0.9368	0.3573	0.2236	0.0912	0.0571	0.0916	0.0601
1,000,000	10	5000	0.9494	0.9455	0.9486	0.9432	0.1600	0.1000	0.0408	0.0255	0.0408	0.0259
1,000,000	10	10,000	0.9495	0.9489	0.9481	0.9445	0.1127	0.0707	0.0288	0.0180	0.0288	0.0182
1,000,000	30	1000	0.9457	0.9205	0.9460	0.9200	0.3567	0.2270	0.0910	0.0579	0.0924	0.0646
1,000,000	30	5000	0.9494	0.9422	0.9482	0.9409	0.1595	0.1014	0.0407	0.0259	0.0409	0.0267
1,000,000	30	10,000	0.9487	0.9458	0.9474	0.9425	0.1133	0.0717	0.0289	0.0183	0.0289	0.0187
1,000,000	50	1000	0.9440	0.9049	0.9440	0.9049	0.3581	0.2273	0.0914	0.0580	0.0945	0.0680
1,000,000	50	5000	0.9484	0.9383	0.9487	0.9373	0.1602	0.1017	0.0409	0.0260	0.0411	0.0271
1,000,000	50	10,000	0.9482	0.9435	0.9470	0.9414	0.1133	0.0719	0.0289	0.0183	0.0291	0.0189

In table (4.15) and (4.16), ε are from T3 and LAP distributions, the coverage probabilities are slightly different from those of the previous tables but the trend and conclusions are the same.

Table 4.16.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim GA, \varepsilon \sim LAP$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9494	0.9361	0.9482	0.9336	0.2927	0.2030	0.0747	0.0518	0.0746	0.0548
100,000	10	5000	0.9506	0.9445	0.9464	0.9357	0.1311	0.0909	0.0334	0.0232	0.0334	0.0236
100,000	10	10,000	0.9494	0.9496	0.9398	0.9276	0.0927	0.0645	0.0237	0.0165	0.0236	0.0165
100,000	30	1000	0.9470	0.9200	0.9457	0.9164	0.2931	0.2066	0.0748	0.0527	0.0755	0.0590
100,000	30	5000	0.9505	0.9393	0.9448	0.9294	0.1314	0.0923	0.0335	0.0235	0.0334	0.0244
100,000	30	10,000	0.9517	0.9462	0.9435	0.9288	0.0930	0.0652	0.0237	0.0166	0.0236	0.0169
100,000	50	1000	0.9452	0.9070	0.9446	0.9045	0.2949	0.2083	0.0752	0.0531	0.0766	0.0620
100,000	50	5000	0.9501	0.9397	0.9469	0.9329	0.1315	0.0931	0.0336	0.0238	0.0334	0.0247
100,000	50	10,000	0.9514	0.9436	0.9424	0.9244	0.0932	0.0654	0.0238	0.0167	0.0237	0.0172
500,000	10	1000	0.9495	0.9363	0.9496	0.9361	0.2924	0.2028	0.0746	0.0517	0.0748	0.0545
500,000	10	5000	0.9504	0.9466	0.9491	0.9442	0.1309	0.0907	0.0334	0.0231	0.0333	0.0235
500,000	10	10,000	0.9495	0.9476	0.9471	0.9414	0.0924	0.0641	0.0236	0.0164	0.0236	0.0165
500,000	30	1000	0.9467	0.9181	0.9470	0.9177	0.2925	0.2056	0.0746	0.0524	0.0757	0.0590
500,000	30	5000	0.9483	0.9426	0.9470	0.9398	0.1307	0.0920	0.0333	0.0235	0.0335	0.0242
500,000	30	10,000	0.9494	0.9452	0.9464	0.9409	0.0924	0.0651	0.0236	0.0166	0.0237	0.0169
500,000	50	1000	0.9438	0.9038	0.9437	0.9024	0.2918	0.2062	0.0744	0.0526	0.0763	0.0621
500,000	50	5000	0.9484	0.9360	0.9470	0.9333	0.1309	0.0921	0.0334	0.0235	0.0336	0.0248
500,000	50	10,000	0.9523	0.9439	0.9498	0.9397	0.0925	0.0653	0.0236	0.0167	0.0235	0.0171
1,000,000	10	1000	0.9463	0.9351	0.9467	0.9348	0.2922	0.2027	0.0746	0.0517	0.0751	0.0548
1,000,000	10	5000	0.9489	0.9477	0.9484	0.9469	0.1306	0.0908	0.0333	0.0232	0.0333	0.0234
1,000,000	10	10,000	0.9490	0.9488	0.9477	0.9459	0.0924	0.0642	0.0236	0.0164	0.0237	0.0164
1,000,000	30	1000	0.9453	0.9189	0.9455	0.9190	0.2922	0.2057	0.0745	0.0525	0.0757	0.0591
1,000,000	30	5000	0.9494	0.9402	0.9493	0.9391	0.1306	0.0919	0.0333	0.0235	0.0333	0.0244
1,000,000	30	10,000	0.9490	0.9451	0.9480	0.9428	0.0924	0.0650	0.0236	0.0166	0.0236	0.0169
1,000,000	50	1000	0.9418	0.9051	0.9414	0.9046	0.2923	0.2061	0.0746	0.0526	0.0769	0.0619
1,000,000	50	5000	0.9494	0.9381	0.9491	0.9374	0.1308	0.0921	0.0334	0.0235	0.0334	0.0247
1,000,000	50	10,000	0.9506	0.9430	0.9495	0.9414	0.0925	0.0652	0.0236	0.0166	0.0236	0.0171

Table (4.17) to table (4.20) show that, when the design matrix is from multivariate log normal distribution, even the coverage probabilities for β_0 and $\hat{\beta}$ based on uniform subsampling could be under and not close to 95% when $\frac{m}{p}$ is less than 500. Under the A-optimal subsampling case, most of the coverage probabilities are close to 95%

except for when m is too small compared to n (for example, $n = 1,000,000, m = 1000$, i.e. $m = 0.001 * n$) or when the number of features p is too large (50), the coverage probabilities for both β_0 and $\hat{\beta}$ could be smaller than the nominal level. This can be explained by that, the A-optimal sampling carries more information from full sample, hence the coverage probabilities could be closer to nominal confidence level if the choice of $\frac{m}{p}$ is appropriate. Our suggestion from this simulation study is that, in order to get a good interval estimate of β_0 and $\hat{\beta}$, the massive data bootstrap sample size m should be greater than 5000, especially when p is large (for example, 50).

Specifically for table (4.17) the length of the A-optimal subsampling confidence interval of the massive data bootstrap regression estimator is shorter than that of the corresponding uniform subsampling method. The standard error of the former is also smaller. Both methods' empirical standard errors are close to theoretical standard errors when $\frac{m}{p}$ is larger than 500.

The results and conclusions of table (4.18) are similar to those of table (4.17).

In table (4.19), when ε is from T3 distribution, the coverage probabilities are slightly different from the previous tables. Larger sample size n and massive data bootstrap subsample size m are needed to get nice result. However, the trend and conclusions are still the same.

In table (4.20), when ε is from LAP distribution, in the case although the sample size is large the choices of massive data bootstrap sample size m in our table are still not large enough. Larger choices of m are needed.

Table 4.17.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim LN, \varepsilon \sim GA$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9005	0.9301	0.8989	0.9257	0.1792	0.0769	0.0457	0.0196	0.0505	0.0197
100,000	10	5000	0.9345	0.9202	0.9293	0.8900	0.0799	0.0321	0.0204	0.0082	0.0197	0.0083
100,000	10	10,000	0.9357	0.9543	0.9275	0.9216	0.0535	0.0244	0.0137	0.0062	0.0133	0.0057
100,000	30	1000	0.9164	0.9127	0.9165	0.9107	0.1862	0.0995	0.0475	0.0254	0.0504	0.0265
100,000	30	5000	0.9393	0.9218	0.9343	0.9015	0.0804	0.0407	0.0205	0.0104	0.0197	0.0107
100,000	30	10,000	0.8977	0.9399	0.8837	0.8997	0.0542	0.0297	0.0138	0.0076	0.0137	0.0070
100,000	50	1000	0.8933	0.8727	0.8925	0.8704	0.1753	0.1014	0.0447	0.0259	0.0506	0.0314
100,000	50	5000	0.9281	0.9385	0.9230	0.9281	0.0783	0.0479	0.0200	0.0122	0.0196	0.0115
100,000	50	10,000	0.9373	0.9247	0.9237	0.8975	0.0591	0.0329	0.0151	0.0084	0.0137	0.0079
500,000	10	1000	0.8803	0.9216	0.8803	0.9206	0.1620	0.0718	0.0413	0.0183	0.0510	0.0196
500,000	10	5000	0.9238	0.9509	0.9230	0.9456	0.0749	0.0322	0.0191	0.0082	0.0198	0.0079
500,000	10	10,000	0.9281	0.9453	0.9259	0.9318	0.0508	0.0222	0.0130	0.0057	0.0136	0.0056
500,000	30	1000	0.8751	0.9035	0.8746	0.9030	0.1575	0.0882	0.0402	0.0225	0.0505	0.0260
500,000	30	5000	0.9191	0.9330	0.9178	0.9279	0.0726	0.0389	0.0185	0.0099	0.0199	0.0101
500,000	30	10,000	0.9433	0.9340	0.9418	0.9256	0.0544	0.0271	0.0139	0.0069	0.0137	0.0070
500,000	50	1000	0.8773	0.8710	0.8773	0.8711	0.1627	0.0943	0.0415	0.0241	0.0517	0.0310
500,000	50	5000	0.9275	0.9241	0.9266	0.9210	0.0746	0.0420	0.0190	0.0107	0.0200	0.0113
500,000	50	10,000	0.9323	0.9290	0.9297	0.9218	0.0521	0.0295	0.0133	0.0075	0.0136	0.0078
1,000,000	10	1000	0.8849	0.9123	0.8848	0.9117	0.1624	0.0684	0.0414	0.0175	0.0504	0.0196
1,000,000	10	5000	0.9280	0.9308	0.9278	0.9281	0.0729	0.0302	0.0186	0.0077	0.0197	0.0079
1,000,000	10	10,000	0.9263	0.9376	0.9248	0.9330	0.0496	0.0215	0.0127	0.0055	0.0135	0.0055
1,000,000	30	1000	0.8843	0.8977	0.8846	0.8973	0.1608	0.0863	0.0410	0.0220	0.0506	0.0263
1,000,000	30	5000	0.9206	0.9402	0.9203	0.9396	0.0718	0.0387	0.0183	0.0099	0.0200	0.0100
1,000,000	30	10,000	0.9367	0.9384	0.9356	0.9338	0.0525	0.0267	0.0134	0.0068	0.0136	0.0069
1,000,000	50	1000	0.8725	0.8852	0.8722	0.8849	0.1589	0.0953	0.0405	0.0243	0.0520	0.0302
1,000,000	50	5000	0.9233	0.9223	0.9228	0.9211	0.0729	0.0415	0.0186	0.0106	0.0200	0.0113
1,000,000	50	10,000	0.9388	0.9345	0.9375	0.9321	0.0518	0.0292	0.0132	0.0075	0.0135	0.0077

Table 4.18.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim LN, \varepsilon \sim LN$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9076	0.9298	0.9073	0.9226	0.3863	0.1220	0.0986	0.0311	0.1106	0.0316
100,000	10	5000	0.9314	0.9219	0.9264	0.8871	0.1669	0.0508	0.0426	0.0130	0.0420	0.0130
100,000	10	10,000	0.9351	0.9561	0.9266	0.8987	0.1113	0.0386	0.0284	0.0098	0.0279	0.0089
100,000	30	1000	0.9208	0.9116	0.9200	0.9068	0.4054	0.1577	0.1034	0.0402	0.1093	0.0423
100,000	30	5000	0.9313	0.9239	0.9275	0.8895	0.1694	0.0643	0.0432	0.0164	0.0420	0.0168
100,000	30	10,000	0.8924	0.9393	0.8818	0.8665	0.1162	0.0470	0.0296	0.0120	0.0299	0.0112
100,000	50	1000	0.8934	0.8694	0.8920	0.8647	0.3610	0.1603	0.0921	0.0409	0.1085	0.0509
100,000	50	5000	0.9216	0.9383	0.9164	0.9200	0.1618	0.0759	0.0413	0.0194	0.0415	0.0183
100,000	50	10,000	0.9317	0.9268	0.9196	0.8826	0.1219	0.0521	0.0311	0.0133	0.0285	0.0125
500,000	10	1000	0.8934	0.9232	0.8926	0.9206	0.3472	0.1136	0.0886	0.0290	0.1085	0.0310
500,000	10	5000	0.9292	0.9494	0.9291	0.9430	0.1616	0.0509	0.0412	0.0130	0.0430	0.0125
500,000	10	10,000	0.9264	0.9446	0.9234	0.9227	0.1073	0.0351	0.0274	0.0090	0.0292	0.0089
500,000	30	1000	0.8855	0.8997	0.8855	0.8990	0.3366	0.1396	0.0859	0.0356	0.1085	0.0421
500,000	30	5000	0.9229	0.9315	0.9218	0.9236	0.1559	0.0615	0.0398	0.0157	0.0433	0.0160
500,000	30	10,000	0.9407	0.9316	0.9376	0.9181	0.1167	0.0429	0.0298	0.0109	0.0294	0.0112
500,000	50	1000	0.8877	0.8711	0.8871	0.8693	0.3504	0.1492	0.0894	0.0381	0.1127	0.0500
500,000	50	5000	0.9265	0.9248	0.9258	0.9188	0.1614	0.0666	0.0412	0.0170	0.0427	0.0179
500,000	50	10,000	0.9317	0.9284	0.9297	0.9158	0.1100	0.0466	0.0281	0.0119	0.0291	0.0125
1,000,000	10	1000	0.8929	0.9126	0.8928	0.9116	0.3394	0.1081	0.0866	0.0276	0.1072	0.0311
1,000,000	10	5000	0.9333	0.9322	0.9331	0.9246	0.1639	0.0478	0.0418	0.0122	0.0426	0.0126
1,000,000	10	10,000	0.9272	0.9383	0.9270	0.9307	0.1070	0.0340	0.0273	0.0087	0.0294	0.0088
1,000,000	30	1000	0.8978	0.8959	0.8977	0.8950	0.3442	0.1366	0.0878	0.0348	0.1096	0.0420
1,000,000	30	5000	0.9246	0.9371	0.9235	0.9328	0.1525	0.0612	0.0389	0.0156	0.0423	0.0160
1,000,000	30	10,000	0.9354	0.9393	0.9342	0.9315	0.1133	0.0423	0.0289	0.0108	0.0291	0.0110
1,000,000	50	1000	0.8794	0.8826	0.8786	0.8817	0.3383	0.1505	0.0863	0.0384	0.1114	0.0489
1,000,000	50	5000	0.9259	0.9230	0.9249	0.9205	0.1564	0.0656	0.0399	0.0167	0.0431	0.0179
1,000,000	50	10,000	0.9352	0.9366	0.9342	0.9313	0.1100	0.0462	0.0281	0.0118	0.0289	0.0122

Table 4.19.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim LN, \varepsilon \sim T3$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9166	0.9152	0.9164	0.9094	0.3122	0.1051	0.0796	0.0268	0.0849	0.0278
100,000	10	5000	0.9151	0.9513	0.9091	0.9135	0.1277	0.0491	0.0326	0.0125	0.0336	0.0112
100,000	10	10,000	0.9066	0.9429	0.8962	0.8692	0.0882	0.0336	0.0225	0.0086	0.0233	0.0079
100,000	30	1000	0.8993	0.8838	0.8984	0.8790	0.3105	0.1293	0.0792	0.0330	0.0903	0.0396
100,000	30	5000	0.9330	0.9285	0.9276	0.9006	0.1403	0.0577	0.0358	0.0147	0.0330	0.0147
100,000	30	10,000	0.9392	0.9429	0.9223	0.8642	0.0928	0.0407	0.0237	0.0104	0.0225	0.0100
100,000	50	1000	0.8969	0.8820	0.8955	0.8769	0.2960	0.1453	0.0755	0.0371	0.0885	0.0446
100,000	50	5000	0.9431	0.9373	0.9397	0.9205	0.1387	0.0648	0.0354	0.0165	0.0334	0.0161
100,000	50	10,000	0.9366	0.9467	0.9290	0.9085	0.0988	0.0460	0.0252	0.0117	0.0241	0.0111
500,000	10	1000	0.8937	0.9109	0.8934	0.9092	0.2812	0.0974	0.0717	0.0248	0.0857	0.0278
500,000	10	5000	0.9239	0.9412	0.9226	0.9310	0.1237	0.0438	0.0316	0.0112	0.0344	0.0111
500,000	10	10,000	0.9377	0.9436	0.9353	0.9229	0.0902	0.0310	0.0230	0.0079	0.0232	0.0077
500,000	30	1000	0.8868	0.8723	0.8859	0.8710	0.2746	0.1166	0.0701	0.0297	0.0866	0.0383
500,000	30	5000	0.9238	0.9330	0.9231	0.9294	0.1235	0.0543	0.0315	0.0139	0.0338	0.0142
500,000	30	10,000	0.9401	0.9413	0.9377	0.9243	0.0915	0.0380	0.0233	0.0097	0.0234	0.0097
500,000	50	1000	0.8788	0.8501	0.8788	0.8492	0.2754	0.1288	0.0702	0.0329	0.0920	0.0451
500,000	50	5000	0.9302	0.9324	0.9299	0.9284	0.1280	0.0591	0.0327	0.0151	0.0344	0.0157
500,000	50	10,000	0.9257	0.9222	0.9242	0.9128	0.0872	0.0400	0.0222	0.0102	0.0227	0.0108
1,000,000	10	1000	0.8778	0.9124	0.8777	0.9110	0.2644	0.0952	0.0675	0.0243	0.0870	0.0274
1,000,000	10	5000	0.9230	0.9320	0.9224	0.9245	0.1228	0.0421	0.0313	0.0108	0.0340	0.0111
1,000,000	10	10,000	0.9361	0.9403	0.9337	0.9287	0.0881	0.0300	0.0225	0.0076	0.0236	0.0077
1,000,000	30	1000	0.8938	0.8696	0.8933	0.8693	0.2799	0.1145	0.0714	0.0292	0.0875	0.0381
1,000,000	30	5000	0.9264	0.9468	0.9263	0.9442	0.1266	0.0551	0.0323	0.0141	0.0345	0.0138
1,000,000	30	10,000	0.9342	0.9293	0.9331	0.9225	0.0876	0.0362	0.0224	0.0093	0.0233	0.0097
1,000,000	50	1000	0.8959	0.8550	0.8958	0.8539	0.2838	0.1271	0.0724	0.0324	0.0926	0.0441
1,000,000	50	5000	0.9266	0.9207	0.9262	0.9178	0.1251	0.0565	0.0319	0.0144	0.0344	0.0159
1,000,000	50	10,000	0.9204	0.9301	0.9194	0.9242	0.0874	0.0406	0.0223	0.0104	0.0234	0.0108

Table 4.20.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim LN, \varepsilon \sim LAP$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.8825	0.8932	0.8818	0.8876	0.2480	0.0931	0.0633	0.0238	0.0719	0.0261
100,000	10	5000	0.9438	0.9598	0.9410	0.9391	0.1172	0.0446	0.0299	0.0114	0.0282	0.0100
100,000	10	10,000	0.9429	0.9322	0.9328	0.8832	0.0773	0.0289	0.0197	0.0074	0.0189	0.0072
100,000	30	1000	0.9185	0.9015	0.9182	0.8974	0.2756	0.1221	0.0703	0.0312	0.0731	0.0351
100,000	30	5000	0.9333	0.9284	0.9269	0.9059	0.1100	0.0527	0.0281	0.0134	0.0278	0.0132
100,000	30	10,000	0.9393	0.9421	0.9322	0.9048	0.0784	0.0382	0.0200	0.0098	0.0192	0.0090
100,000	50	1000	0.8972	0.8854	0.8963	0.8819	0.2518	0.1334	0.0642	0.0340	0.0723	0.0408
100,000	50	5000	0.9426	0.8970	0.9389	0.8806	0.1145	0.0537	0.0292	0.0137	0.0279	0.0151
100,000	50	10,000	0.9289	0.9323	0.9208	0.9090	0.0767	0.0409	0.0196	0.0104	0.0192	0.0101
500,000	10	1000	0.8975	0.9056	0.8971	0.9048	0.2421	0.0876	0.0618	0.0224	0.0720	0.0254
500,000	10	5000	0.9327	0.9405	0.9310	0.9324	0.1063	0.0392	0.0271	0.0100	0.0279	0.0100
500,000	10	10,000	0.9353	0.9259	0.9337	0.9113	0.0735	0.0268	0.0188	0.0068	0.0191	0.0071
500,000	30	1000	0.8877	0.8760	0.8874	0.8750	0.2365	0.1073	0.0603	0.0274	0.0729	0.0350
500,000	30	5000	0.9131	0.9234	0.9124	0.9201	0.1027	0.0481	0.0262	0.0123	0.0281	0.0130
500,000	30	10,000	0.9366	0.9409	0.9345	0.9313	0.0748	0.0347	0.0191	0.0089	0.0192	0.0088
500,000	50	1000	0.8808	0.8576	0.8810	0.8564	0.2319	0.1175	0.0592	0.0300	0.0728	0.0405
500,000	50	5000	0.9183	0.9216	0.9177	0.9180	0.1026	0.0535	0.0262	0.0137	0.0282	0.0145
500,000	50	10,000	0.9293	0.9304	0.9258	0.9215	0.0721	0.0371	0.0184	0.0095	0.0191	0.0098
1,000,000	10	1000	0.8846	0.9000	0.8843	0.9001	0.2354	0.0846	0.0600	0.0216	0.0724	0.0251
1,000,000	10	5000	0.8963	0.9170	0.8964	0.9148	0.0969	0.0370	0.0247	0.0094	0.0281	0.0102
1,000,000	10	10,000	0.9159	0.9269	0.9141	0.9198	0.0695	0.0265	0.0177	0.0068	0.0192	0.0070
1,000,000	30	1000	0.8825	0.8700	0.8824	0.8697	0.2311	0.1061	0.0590	0.0271	0.0724	0.0348
1,000,000	30	5000	0.9235	0.9239	0.9234	0.9214	0.1024	0.0471	0.0261	0.0120	0.0281	0.0129
1,000,000	30	10,000	0.9324	0.9215	0.9315	0.9165	0.0735	0.0324	0.0187	0.0083	0.0193	0.0089
1,000,000	50	1000	0.8809	0.8575	0.8806	0.8566	0.2287	0.1159	0.0583	0.0296	0.0726	0.0401
1,000,000	50	5000	0.9210	0.9278	0.9206	0.9265	0.1015	0.0524	0.0259	0.0134	0.0281	0.0143
1,000,000	50	10,000	0.9384	0.9210	0.9380	0.9186	0.0736	0.0360	0.0188	0.0092	0.0192	0.0099

For table (4.21) to table (4.24), the design matrix is from T3 distribution which has heavier tails compared to normal distribution. The results are similar to those of the multivariate Gaussian distribution (i.e. tables (4.13)-(4.16)). We need an appropriate choice of m while taking into consideration values of n and p. Generally, if n or p gets large then a larger m is needed in order to achieve the nominal confidence level.

Table 4.21.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim T3, \varepsilon \sim GA$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9379	0.9395	0.9363	0.9363	0.2127	0.1374	0.0543	0.0351	0.0564	0.0362
100,000	10	5000	0.9465	0.9493	0.9403	0.9356	0.0954	0.0608	0.0243	0.0155	0.0245	0.0153
100,000	10	10,000	0.9544	0.9520	0.9408	0.9166	0.0685	0.0426	0.0175	0.0109	0.0170	0.0107
100,000	30	1000	0.9366	0.9263	0.9358	0.9246	0.2204	0.1427	0.0562	0.0364	0.0590	0.0395
100,000	30	5000	0.9554	0.9515	0.9503	0.9398	0.1016	0.0646	0.0259	0.0165	0.0251	0.0162
100,000	30	10,000	0.9561	0.9536	0.9436	0.9244	0.0706	0.0445	0.0180	0.0114	0.0173	0.0111
100,000	50	1000	0.9313	0.9156	0.9306	0.9130	0.2256	0.1481	0.0576	0.0378	0.0616	0.0428
100,000	50	5000	0.9558	0.9568	0.9519	0.9479	0.1020	0.0665	0.0260	0.0170	0.0251	0.0164
100,000	50	10,000	0.9579	0.9590	0.9528	0.9461	0.0715	0.0469	0.0182	0.0120	0.0175	0.0114
500,000	10	1000	0.9340	0.9319	0.9338	0.9324	0.2110	0.1321	0.0538	0.0337	0.0568	0.0360
500,000	10	5000	0.9478	0.9487	0.9462	0.9461	0.0943	0.0593	0.0241	0.0151	0.0242	0.0152
500,000	10	10,000	0.9476	0.9522	0.9451	0.9456	0.0666	0.0419	0.0170	0.0107	0.0170	0.0105
500,000	30	1000	0.9258	0.9108	0.9252	0.9104	0.2139	0.1358	0.0546	0.0346	0.0597	0.0400
500,000	30	5000	0.9421	0.9426	0.9412	0.9394	0.0944	0.0605	0.0241	0.0154	0.0247	0.0159
500,000	30	10,000	0.9481	0.9476	0.9449	0.9415	0.0679	0.0432	0.0173	0.0110	0.0174	0.0111
500,000	50	1000	0.9162	0.8988	0.9161	0.8985	0.2149	0.1384	0.0548	0.0353	0.0619	0.0424
500,000	50	5000	0.9432	0.9404	0.9420	0.9376	0.0956	0.0614	0.0244	0.0157	0.0251	0.0163
500,000	50	10,000	0.9489	0.9479	0.9464	0.9424	0.0684	0.0440	0.0174	0.0112	0.0174	0.0113
1,000,000	10	1000	0.9293	0.9327	0.9293	0.9321	0.2070	0.1315	0.0528	0.0336	0.0569	0.0359
1,000,000	10	5000	0.9445	0.9464	0.9433	0.9445	0.0931	0.0587	0.0238	0.0150	0.0243	0.0152
1,000,000	10	10,000	0.9473	0.9463	0.9457	0.9440	0.0660	0.0413	0.0168	0.0105	0.0171	0.0106
1,000,000	30	1000	0.9242	0.9093	0.9242	0.9096	0.2115	0.1347	0.0540	0.0344	0.0595	0.0397
1,000,000	30	5000	0.9391	0.9406	0.9383	0.9393	0.0936	0.0600	0.0239	0.0153	0.0249	0.0159
1,000,000	30	10,000	0.9456	0.9462	0.9455	0.9434	0.0671	0.0428	0.0171	0.0109	0.0173	0.0111
1,000,000	50	1000	0.9129	0.8878	0.9130	0.8873	0.2142	0.1377	0.0547	0.0351	0.0624	0.0433
1,000,000	50	5000	0.9404	0.9354	0.9406	0.9343	0.0950	0.0608	0.0242	0.0155	0.0251	0.0164
1,000,000	50	10,000	0.9477	0.9456	0.9462	0.9438	0.0683	0.0438	0.0174	0.0112	0.0176	0.0113

Table 4.22.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim T3, \varepsilon \sim LN$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9377	0.9402	0.9367	0.9339	0.4647	0.2175	0.1186	0.0555	0.1215	0.0570
100,000	10	5000	0.9451	0.9497	0.9395	0.9256	0.2062	0.0963	0.0526	0.0246	0.0530	0.0242
100,000	10	10,000	0.9529	0.9498	0.9430	0.8993	0.1472	0.0675	0.0376	0.0172	0.0365	0.0169
100,000	30	1000	0.9364	0.9259	0.9345	0.9200	0.4826	0.2260	0.1231	0.0577	0.1290	0.0630
100,000	30	5000	0.9533	0.9498	0.9482	0.9263	0.2183	0.1024	0.0557	0.0261	0.0542	0.0258
100,000	30	10,000	0.9541	0.9510	0.9450	0.9026	0.1543	0.0705	0.0394	0.0180	0.0379	0.0177
100,000	50	1000	0.9316	0.9131	0.9307	0.9072	0.4853	0.2342	0.1238	0.0598	0.1327	0.0683
100,000	50	5000	0.9524	0.9545	0.9480	0.9350	0.2173	0.1053	0.0554	0.0269	0.0541	0.0262
100,000	50	10,000	0.9575	0.9612	0.9503	0.9263	0.1534	0.0742	0.0391	0.0189	0.0376	0.0179
500,000	10	1000	0.9328	0.9313	0.9327	0.9304	0.4526	0.2092	0.1155	0.0534	0.1222	0.0571
500,000	10	5000	0.9454	0.9486	0.9447	0.9447	0.2024	0.0939	0.0516	0.0240	0.0523	0.0240
500,000	10	10,000	0.9481	0.9540	0.9450	0.9391	0.1471	0.0663	0.0375	0.0169	0.0367	0.0166
500,000	30	1000	0.9262	0.9102	0.9262	0.9089	0.4602	0.2148	0.1174	0.0548	0.1284	0.0631
500,000	30	5000	0.9434	0.9440	0.9418	0.9391	0.2040	0.0958	0.0520	0.0244	0.0533	0.0251
500,000	30	10,000	0.9478	0.9472	0.9455	0.9362	0.1465	0.0683	0.0374	0.0174	0.0375	0.0175
500,000	50	1000	0.9187	0.8931	0.9181	0.8921	0.4646	0.2191	0.1185	0.0559	0.1339	0.0679
500,000	50	5000	0.9408	0.9394	0.9409	0.9367	0.2050	0.0972	0.0523	0.0248	0.0540	0.0259
500,000	50	10,000	0.9473	0.9509	0.9451	0.9424	0.1479	0.0697	0.0377	0.0178	0.0379	0.0177
1,000,000	10	1000	0.9299	0.9315	0.9297	0.9317	0.4435	0.2081	0.1131	0.0531	0.1220	0.0568
1,000,000	10	5000	0.9447	0.9474	0.9438	0.9436	0.2038	0.0929	0.0520	0.0237	0.0523	0.0239
1,000,000	10	10,000	0.9449	0.9474	0.9446	0.9431	0.1413	0.0653	0.0360	0.0167	0.0365	0.0168
1,000,000	30	1000	0.9231	0.9105	0.9232	0.9096	0.4559	0.2132	0.1163	0.0544	0.1291	0.0630
1,000,000	30	5000	0.9394	0.9403	0.9387	0.9381	0.2012	0.0949	0.0513	0.0242	0.0534	0.0251
1,000,000	30	10,000	0.9461	0.9486	0.9448	0.9434	0.1453	0.0678	0.0371	0.0173	0.0371	0.0174
1,000,000	50	1000	0.9132	0.8856	0.9131	0.8854	0.4611	0.2179	0.1176	0.0556	0.1355	0.0687
1,000,000	50	5000	0.9402	0.9373	0.9403	0.9338	0.2064	0.0963	0.0526	0.0246	0.0546	0.0258
1,000,000	50	10,000	0.9452	0.9460	0.9443	0.9413	0.1474	0.0694	0.0376	0.0177	0.0382	0.0179

Table 4.23.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim T3, \varepsilon \sim T3$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9403	0.9351	0.9394	0.9298	0.3751	0.1884	0.0957	0.0481	0.0991	0.0507
100,000	10	5000	0.9486	0.9499	0.9413	0.9219	0.1631	0.0835	0.0416	0.0213	0.0414	0.0211
100,000	10	10,000	0.9509	0.9532	0.9422	0.9154	0.1178	0.0600	0.0301	0.0153	0.0295	0.0149
100,000	30	1000	0.9375	0.9114	0.9367	0.9063	0.3829	0.1950	0.0977	0.0498	0.1016	0.0570
100,000	30	5000	0.9501	0.9521	0.9449	0.9298	0.1699	0.0887	0.0434	0.0226	0.0428	0.0223
100,000	30	10,000	0.9556	0.9546	0.9450	0.9129	0.1188	0.0622	0.0303	0.0159	0.0294	0.0155
100,000	50	1000	0.9327	0.9043	0.9320	0.9003	0.3911	0.2035	0.0998	0.0519	0.1063	0.0611
100,000	50	5000	0.9527	0.9471	0.9472	0.9281	0.1745	0.0907	0.0445	0.0231	0.0436	0.0233
100,000	50	10,000	0.9575	0.9527	0.9496	0.9278	0.1228	0.0639	0.0313	0.0163	0.0300	0.0159
500,000	10	1000	0.9336	0.9298	0.9331	0.9290	0.3576	0.1820	0.0912	0.0464	0.0974	0.0502
500,000	10	5000	0.9475	0.9447	0.9468	0.9407	0.1636	0.0811	0.0417	0.0207	0.0417	0.0210
500,000	10	10,000	0.9439	0.9473	0.9416	0.9401	0.1144	0.0573	0.0292	0.0146	0.0296	0.0148
500,000	30	1000	0.9281	0.9024	0.9282	0.9015	0.3674	0.1874	0.0937	0.0478	0.1032	0.0566
500,000	30	5000	0.9442	0.9422	0.9434	0.9383	0.1642	0.0839	0.0419	0.0214	0.0426	0.0221
500,000	30	10,000	0.9451	0.9446	0.9443	0.9378	0.1159	0.0590	0.0296	0.0151	0.0300	0.0154
500,000	50	1000	0.9216	0.8841	0.9218	0.8830	0.3707	0.1909	0.0946	0.0487	0.1062	0.0607
500,000	50	5000	0.9433	0.9331	0.9423	0.9286	0.1678	0.0848	0.0428	0.0216	0.0439	0.0231
500,000	50	10,000	0.9483	0.9455	0.9444	0.9358	0.1176	0.0597	0.0300	0.0152	0.0300	0.0155
1,000,000	10	1000	0.9366	0.9261	0.9364	0.9259	0.3605	0.1805	0.0920	0.0460	0.0973	0.0503
1,000,000	10	5000	0.9440	0.9447	0.9435	0.9412	0.1602	0.0807	0.0409	0.0206	0.0417	0.0210
1,000,000	10	10,000	0.9468	0.9527	0.9455	0.9484	0.1150	0.0583	0.0293	0.0149	0.0296	0.0147
1,000,000	30	1000	0.9257	0.9019	0.9255	0.9018	0.3660	0.1861	0.0934	0.0475	0.1039	0.0562
1,000,000	30	5000	0.9419	0.9371	0.9413	0.9353	0.1632	0.0834	0.0416	0.0213	0.0431	0.0223
1,000,000	30	10,000	0.9440	0.9469	0.9427	0.9406	0.1148	0.0590	0.0293	0.0151	0.0300	0.0152
1,000,000	50	1000	0.9197	0.8783	0.9197	0.8775	0.3850	0.1892	0.0982	0.0483	0.1064	0.0612
1,000,000	50	5000	0.9410	0.9356	0.9410	0.9332	0.1653	0.0845	0.0422	0.0216	0.0436	0.0228
1,000,000	50	10,000	0.9467	0.9431	0.9455	0.9376	0.1173	0.0596	0.0299	0.0152	0.0303	0.0156

Table 4.24.: Massive Data Bootstrapping Confidence Interval Comparison, $\mathbf{x} \sim T3, \varepsilon \sim LAP$

n	p	m	CP $\hat{\beta}$		CP β_0		len		$\hat{se}$.th		$\hat{se}$.data	
			Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt	Unif	Opt
100,000	10	1000	0.9395	0.9349	0.9380	0.9324	0.3026	0.1712	0.0772	0.0437	0.0797	0.0458
100,000	10	5000	0.9531	0.9480	0.9473	0.9270	0.1373	0.0757	0.0350	0.0193	0.0342	0.0192
100,000	10	10,000	0.9547	0.9530	0.9442	0.9212	0.0963	0.0538	0.0246	0.0137	0.0239	0.0134
100,000	30	1000	0.9353	0.9125	0.9343	0.9073	0.3136	0.1784	0.0800	0.0455	0.0841	0.0520
100,000	30	5000	0.9524	0.9481	0.9460	0.9290	0.1410	0.0798	0.0360	0.0204	0.0351	0.0204
100,000	30	10,000	0.9578	0.9550	0.9461	0.9200	0.1004	0.0568	0.0256	0.0145	0.0246	0.0141
100,000	50	1000	0.9318	0.8971	0.9309	0.8931	0.3191	0.1824	0.0814	0.0465	0.0870	0.0558
100,000	50	5000	0.9542	0.9468	0.9486	0.9312	0.1436	0.0825	0.0366	0.0211	0.0356	0.0212
100,000	50	10,000	0.9600	0.9576	0.9505	0.9304	0.1021	0.0588	0.0261	0.0150	0.0247	0.0144
500,000	10	1000	0.9378	0.9253	0.9376	0.9241	0.2981	0.1662	0.0760	0.0424	0.0795	0.0463
500,000	10	5000	0.9472	0.9443	0.9459	0.9405	0.1343	0.0742	0.0343	0.0189	0.0345	0.0193
500,000	10	10,000	0.9443	0.9433	0.9418	0.9369	0.0931	0.0520	0.0238	0.0133	0.0241	0.0135
500,000	30	1000	0.9243	0.9017	0.9247	0.9009	0.2988	0.1709	0.0762	0.0436	0.0837	0.0516
500,000	30	5000	0.9468	0.9410	0.9466	0.9376	0.1363	0.0767	0.0348	0.0196	0.0351	0.0202
500,000	30	10,000	0.9503	0.9487	0.9476	0.9416	0.0964	0.0544	0.0246	0.0139	0.0245	0.0139
500,000	50	1000	0.9167	0.8767	0.9165	0.8764	0.3029	0.1718	0.0773	0.0438	0.0869	0.0557
500,000	50	5000	0.9436	0.9370	0.9425	0.9322	0.1365	0.0771	0.0348	0.0197	0.0356	0.0208
500,000	50	10,000	0.9502	0.9437	0.9475	0.9337	0.0967	0.0548	0.0247	0.0140	0.0246	0.0143
1,000,000	10	1000	0.9340	0.9236	0.9339	0.9236	0.2948	0.1645	0.0752	0.0420	0.0798	0.0462
1,000,000	10	5000	0.9452	0.9447	0.9452	0.9429	0.1330	0.0735	0.0339	0.0187	0.0344	0.0192
1,000,000	10	10,000	0.9456	0.9472	0.9452	0.9450	0.0929	0.0519	0.0237	0.0132	0.0239	0.0134
1,000,000	30	1000	0.9276	0.8963	0.9274	0.8958	0.3008	0.1685	0.0767	0.0430	0.0838	0.0517
1,000,000	30	5000	0.9437	0.9382	0.9429	0.9373	0.1345	0.0758	0.0343	0.0193	0.0351	0.0202
1,000,000	30	10,000	0.9466	0.9458	0.9452	0.9419	0.0945	0.0535	0.0241	0.0137	0.0244	0.0138
1,000,000	50	1000	0.9168	0.8752	0.9165	0.8747	0.3004	0.1711	0.0766	0.0437	0.0867	0.0559
1,000,000	50	5000	0.9413	0.9330	0.9408	0.9326	0.1350	0.0762	0.0344	0.0195	0.0357	0.0209
1,000,000	50	10,000	0.9441	0.9423	0.9422	0.9379	0.0948	0.0540	0.0242	0.0138	0.0247	0.0143

The simulation results in this section support our theoretical findings, and show that for massive data the proposed massive data bootstrapping via A-optimal subsampling works better than uniform subsampling method.

4.2.2 Running Time Comparison

Running time is compared and the results are shown in table (4.25). The choice of n vary in 1,000,000, 2,000,000 and 3,000,000. The value of p varies among 10, 50, 100 and 200. The case with largest $n \times p$ is $n = 3,000,000$ and $p = 200$. In this case, the size of design matrix is right under the limit of the memory of my office computer.

The massive data bootstrap sample size m vary in $0.01n$, $0.05n$ to $0.1n$. The full sample running time and subsample running time are given in the table. As can be seen, within each n and p combination the running time of subsampling method is shorter than that of the full sample running time, and changes according to the choice of m. The gain (saved running time) of proposed subsampling method becomes more evident when n or p increases.

Table 4.25.: Massive Data Bootstrapping Computing Time Comparison

n	p	MDB Sample Running Time			Full Sample Running Time
		$m = 0.01n$	$m = 0.05n$	$m = 0.1n$	
1,000,000	10	1.24	1.36	1.38	1.35
1,000,000	50	3.96	4.59	4.72	6.00
1,000,000	100	8.33	8.99	9.77	17.07
1,000,000	200	19.18	21.48	23.71	54.69
2,000,000	10	2.50	2.91	2.92	2.82
2,000,000	50	7.93	8.55	9.87	13.95
2,000,000	100	16.40	17.25	18.89	41.56
2,000,000	200	36.75	42.99	48.14	140.59
3,000,000	10	3.52	3.88	3.88	3.95
3,000,000	50	10.89	11.41	12.75	22.51
3,000,000	100	21.81	25.67	26.52	62.65
3,000,000	200	56.07	62.64	75.65	203.07

5. REAL DATA APPLICATION IN NATURAL LANGUAGE PROCESSING

In Natural Language Processing (NLP), clustering analysis is an important topic in different types of problems. For example, grouping documents into different topics, clustering words into different categories. In this analysis, we focus on the word level clustering. Grouping words and finding the word similarities are useful for further NLP tasks. We apply our k-means via A-optimal subsampling algorithm to group the Word2Vec embedded word vectors.

Word2Vec is an algorithm developed by Google. Using two layers of neural networks, The input of Word2Vec is a large structured set of texts, the output is a space of vector representation of words. This output word vector file could be used as input matrix in natural language processing and machine learning models. One way to investigate the output vector representations is to find the closest words for a pre-specified word using the distance between word vectors. The other way is to perform k-means clustering on the word vectors to find word classes on huge data sets.

5.1 One Billion Word Benchmark Data

Ciprian Chelba, *et al* (2014) proposed a corpus which includes almost one billion words from WMT 2011 News Crawl data. This data consist of 100 txt files. Each txt file consists of different number of sentences from news, but each file is around 40 MB in memory size. In our study, due to the limited resources, we use the first ten txt files which in total contains around one hundred million words. The following text data preprocessing steps were done in Python3.7 using the NLTK package before the clustering analysis. The Natural Language Tool Kit (NLTK) is a popular package in Python for text analysis.

Firstly, the text data are tokenized into a list of sentences. Then each sentence is tokenized into a list of words. Stop words such as "a, an, the", etc. are removed. Numbers and punctuation marks are also removed. Then the words are embedded into vectors through the Word2Vec function in the gensim package in Python3.7.

The dimension of each word vector is an input of Word2Vec function, here we choose $d = 50$ and $d = 100$. By excluding the words that appear too few times we can control the number of different words n. Here we choose to exclude words that appear 5 or fewer times for $d = 100$, and 10 or fewer times for $d = 50$. As a result, we get $n = 133,386$ in the former, and $n = 90,636$ in the later case. So the combinations of n, d in this study are $(n = 133,386, d = 100)$ and $(n = 90,636, d = 50)$.

The selection of number of clusters k is still a ongoing research question in the area of k-means clustering. However, it is not our research focus. In our case, we assume it is either given by the professionals or automatically chosen by algorithms like elbow method, gap statistics method, etc. We perform the elbow method here to find the optimal k.

As can be seen from figure (5.1), the elbow (where the curve has a large changing angle)is around 6 to 12. So we will perform the A-optimal k-means clustering with $k = 6, 8, 10, 12$.

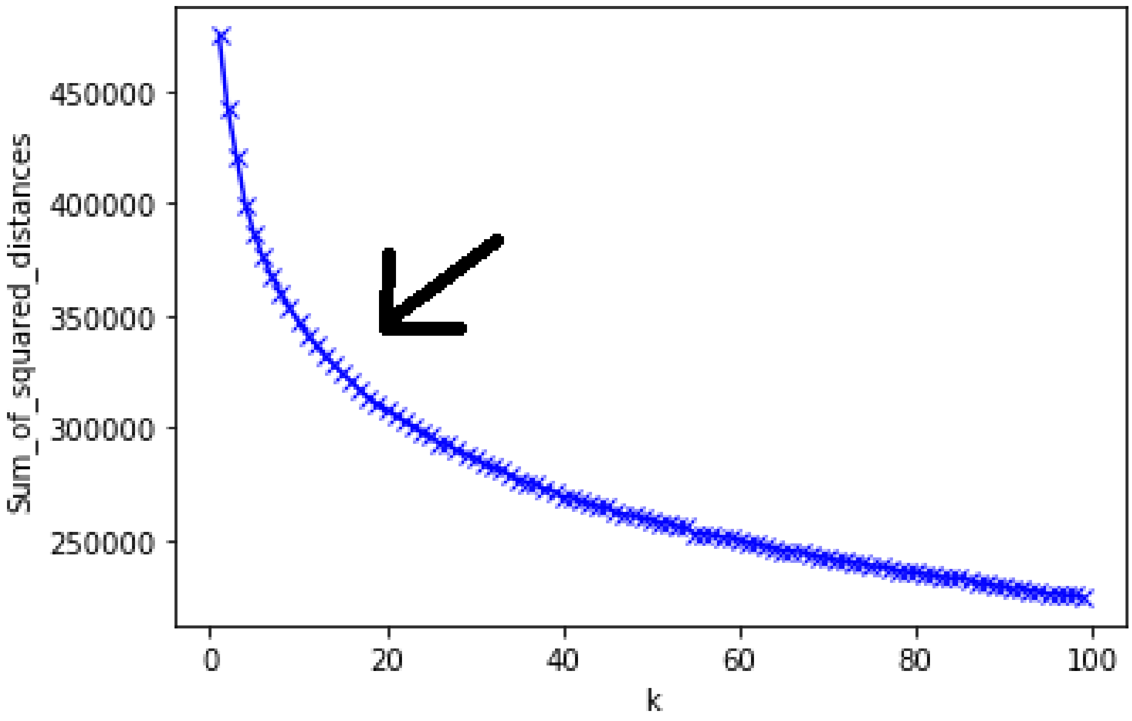


Figure 5.1.: Elbow Method For Optimal k

Similar to the simulation study, the MSE and TimeRatio are compared in different scenarios in table (5.1).

In addition, V-Measure, a widely used clustering evaluation measure in machine learning area based on entropy, is also compared. The measure is written into the Scikit-learn package in Python3.7 and widely used by data scientists. The V-Measure is calculated as

$$v = \frac{(1+\beta) * h * c}{(\beta * h + c)},$$

in which h is the homogeneity and c is the completeness. "h is maximized when each cluster contains elements of as few different classes as possible. c aims to put all elements of each class in single clusters"- more details can be found in RosenBerg (2007). It evaluates how similar two clustering results are. Here, we apply the V-Measure to evaluate how close the k-means clustering via subsampling is to the full sample k-means clustering. Larger V-Measure value indicates higher similarity of the two clustering results.

The choice of r and r_0 could also be a research topic that is worth further investigation. Here we choose combinations $(\frac{r}{n}, \frac{r_0}{n}) = (0.1, 0.1),\ (0.1, 0.05)$ and $(0.05, 0.05)$.

Table 5.1.: One Billion Word Benchmark Data Analysis, n=133,386, p=100

k	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		V-Measure		TimeRatio	
			Unif	Opt	Unif	Opt	Unif	Opt
6	0.1	0.1	13.28735	10.43175	0.73010	0.75839	0.06592	0.09816
6	0.1	0.05	13.39810	11.88511	0.72938	0.75969	0.06731	0.09650
6	0.05	0.05	71.29101	22.10437	0.63548	0.66704	0.03024	0.05413
8	0.1	0.1	28.75562	22.03969	0.69926	0.72883	0.06539	0.11145
8	0.1	0.05	33.88225	26.71326	0.69153	0.71052	0.06565	0.10549
8	0.05	0.05	162.33309	101.52292	0.61129	0.67721	0.03169	0.05669
10	0.1	0.1	133.48044	31.30001	0.66873	0.73680	0.06200	0.10207
10	0.1	0.05	119.36602	28.71909	0.68038	0.73252	0.06212	0.10869
10	0.05	0.05	473.63049	184.53649	0.60326	0.67374	0.02592	0.04870
12	0.1	0.1	247.04930	171.84341	0.64947	0.67873	0.07049	0.12572
12	0.1	0.05	207.34528	100.25191	0.64717	0.65472	0.07096	0.14888
12	0.05	0.05	603.00740	479.73410	0.60357	0.61900	0.03390	0.06285

Table 5.2.: One Billion Word Benchmark Data Analysis, n=90,636, p=50

k	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		V-Measure		TimeRatio	
			Unif	Opt	Unif	Opt	Unif	Opt
6	0.1	0.1	24.97588	21.23650	0.70092	0.73413	0.05542	0.07572
6	0.1	0.05	21.83849	20.50215	0.69058	0.69050	0.05385	0.07481
6	0.05	0.05	209.98194	75.18802	0.59624	0.64431	0.02517	0.03985
8	0.1	0.1	109.14436	21.94016	0.65127	0.69814	0.04880	0.07336
8	0.1	0.05	125.01828	60.67937	0.65109	0.66635	0.04950	0.07575
8	0.05	0.05	575.60486	72.86403	0.57675	0.61025	0.01965	0.03220
10	0.1	0.1	293.39433	328.22987	0.63814	0.65493	0.05849	0.09193
10	0.1	0.05	290.18361	297.08388	0.62949	0.63696	0.05821	0.08883
10	0.05	0.05	766.24854	140.43972	0.56934	0.60905	0.02310	0.03919
12	0.1	0.1	402.13529	106.67088	0.64180	0.66857	0.05536	0.09704
12	0.1	0.05	451.66719	151.64070	0.63925	0.65696	0.05222	0.08805
12	0.05	0.05	987.79380	376.29807	0.57528	0.61078	0.02084	0.03874

From table (5.1) and (5.2) we can see, under different scenarios, except for one or two special cases, the MSE of A-optimal subsampling is always smaller than that of uniform subsampling. The V-Measure of A-optimal subsampling being larger than that of uniform subsampling tells us that the A-optimal subsampling result is closer to the full sample result in this massive data example. The TimeRatio term columns indicate A-optimal subsampling method takes more but reasonable time. When r is smaller, the time ratio becomes smaller.

5.2 Google Word2Vec Data

In this section we apply k-means clustering via A-optimal subsampling to Google's trained Word2Vec word vectors and compare with its performance to that of the uniform subsampling method.

Google published a pre-trained vectors of Google News data set which includes about 100 billion words. The model contains 3 million different words, each word is represented by a 300-dimensional vector. So $n = 3000000$, $p = 300$. True number of clusters k is unknown from our experience that the number of topics in news is normally less than 20, for example, politics, sports, holidays, etc, and also based on our analysis of One Billion Word Benchmark data, we also choose $k = 6, 8, 10, 12$. Also we choose the following combinations of r and r_0: (0.05n,0.01n), (0.01n,0.01n) and (0.01n, 0.005n). The output comparing MSE, V-Measure and time ratio is shown in table (5.3).

Table 5.3.: Google Word2Vec Data Analysis

k	$\frac{r}{n}$	$\frac{r_0}{n}$	MSE		V-Measure		TimeRatio	
			Unif	Opt	Unif	Opt	Unif	Opt
6	0.05	0.01	0.01660	0.01003	0.93473	0.94194	0.06294	0.07300
6	0.01	0.01	0.05221	0.02578	0.87658	0.89023	0.01519	0.01828
6	0.01	0.005	0.05258	0.04220	0.87243	0.87815	0.01526	0.01736
8	0.05	0.01	0.26066	0.17668	0.86376	0.88786	0.05197	0.06276
8	0.01	0.01	0.43432	0.14358	0.80822	0.85351	0.01202	0.01481
8	0.01	0.005	0.40260	0.23743	0.80474	0.83254	0.01177	0.01380
10	0.05	0.01	0.09017	0.05717	0.90163	0.91549	0.04638	0.05517
10	0.01	0.01	0.64623	0.25518	0.77702	0.82814	0.00997	0.01259
10	0.01	0.005	0.36848	0.32515	0.78095	0.81395	0.00996	0.01362
12	0.05	0.01	0.32397	0.30562	0.80525	0.81311	0.04083	0.04911
12	0.01	0.01	1.39106	0.46060	0.71586	0.74465	0.00786	0.01047
12	0.01	0.005	3.75142	1.71544	0.69795	0.75010	0.00858	0.01056

From the table we can see, for different k and combinations of r and r_0, the MSE of the centroid estimator from A-optimal subsampling is always smaller than that of uniform subsampling. By comparing the V-Measure we can see the A-optimal subsampling method has higher V-Measure values, hence its clustering results are closer to those of the full sample compared to the uniform subsampling method.. As for time ratio, the computation times of the A-optimal subsampling method are longer but acceptable. Although the difference in computation time becomes larger when the number of clusters k gets larger, the computing time of proposed method is still acceptable. In conclusion, the A-optimal subsampling outperforms the uniform subsampling in the k-means analysis of this Google Word2Vec real data not only in MSE, but also in V-Measure while the computational time cost is comparable.

REFERENCES

REFERENCES

[1] Avron, H., Maymounkov, P. and Toledo, S. (2010). Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, **32**: 1217–1236.

[2] Bickel,P.J.,Gotze, F.,Zwet, W.R.van (1997). Resampling fewer than n observations: gains, losses, and remedies for losses. *Statistica Sinica*, **7**:(1997):1-31.

[3] Bickel,P.J.,Sakov, A. (2008). On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statistica Sinica*, **7**:(2008):967-985.

[4] Bickel,P.J., Freedman, D.A. (1981). Some asymptotic theory for the bootstrap. *The Annals of Statistics*, **9**(6): 1196-1217.

[5] Bock, H.H. (2008). Origins and extensions of the k-means algorithm in cluster analysis. *Electronic Journ@l for History of Probability and Statistics*, **4**(2).

[6] Boutsidis, C., Mahoney, M.W. and Drineas. P. (2009). An improved approximation algorithm for the column subset selection problem. *In Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*: p. 968–977.

[7] Chelba, C., Mikolov, T.,Schuster, M.,Ge, Q., Brants, T., Koehn, P., Robinson, T. (2014). One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. *arXiv:1312.3005*.

[8] Chatterjee, S. and Bose, A. (2002). Dimension asymptotics for generalized bootstrap in linear regression. *Ann. Inst. Statist. Math.* **54** (2): 367–381.

[9] Peng, H. and Tan, F. (2018). A-optimal Subsampling For Big Data General Estimating Equations. *Manuscript*. Available at `https://www.math.iupui.edu/~hpeng/preprints_hp.html`.

[10] DiCiccio, T.J., Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science*, **11**(3): 189-228.

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

[12] Drineas P., Kannan R. and Mahoney M.W. (2006a). Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, **36**: 132–157.

[13] Drineas, P., Mahoney,M.W., Muthukrishnan, S. and Sarlós, T. (2010). Faster least squares approximation. *Numerische Mathematik*, **117**(2): 219–249.

[14] Drineas P., Mahoney M.W. and Muthukrishnan S. (2006b). Sampling algorithms for ℓ_2 regression and applications. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1127–1136.

[15] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, **7**:(1):1-26.

[16] Efron, B., (1987). Better bootstrap confidence intervals. *Journal of the American Statistical Association*, **82**(397): 171-185.

[17] Efron, B., Tibshirani, R.(1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, **1**(1):54-77.

[18] Eicker, F., Gotze, F., Zwet W. R.van (1963). Asymptotic Normality and Consistency of the Least Squares Estimators for Families of Linear Regressions. *The Annals of Mathematical Statistics*, **34**(2): 447-456.

[19] Fisher, W.D. (1958). On Grouping for Maximum Homogeneity. *The Annals of Probability*, **53**(284): 789-798.

[20] Fan, J., Han, F. and Liu, H. (2013). Challenges of big data analysis. *arXiv:1308.1479.*

[21] Freedman, D.A. (1981). Bootstrapping regression models. *Ann. Statist.* **9**(6): 1218–1228.

[22] Kunsch, H.R. (1989). Bootstrap for general stationary observations. *The Annals of Statistics*, **17**(3): 1217-1241.

[23] Lloyd, Stuart P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, **28**(2):129-137.doi:10.1109/TIT.1982.1056489..

[24] Ma, P. and Sun, X. (2014). Leveraging for big data regression. *Computational Statistics.* textbf7 (1): 70-76.

[25] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, **1**: 281-297.

[26] Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *arXiv:1104.5557v3* [cs.DS]

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

[28] Mikolov, T.,Sutskever, I.,Chen, K., Corrado, G., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv:1310.4546.*

[29] Præstgaard, J. and Wellner, J. A. (1993). Exchangeably weighted bootstraps of the general empirical process. *Ann. Probab.*, **21** (4): 2053–2086.

[30] Peng, H. and Tan, F. (2018a). A Fast Algorithm For Computing The A-optimal Sampling Distributions In Big Data Linear Regression. *Preprint.* Available at `https://www.math.iupui.edu/~hpeng/preprints_hp.html`.

[31] Peng, H. and Tan, F. (2018b). Big Data Linear Regression Via A-optimal Subsampling. Submitted to *Ann. Statist.* Available at `https://www.math.iupui.edu/~hpeng/preprints_hp.html`.

[32] Pollard, D. (1981). Strong consistency of k-means clustering. *The Annals of Statistics*, **9**(1): 135-140.

[33] Pollard, D. (1982). A central limit theorem for k-means clustering. *The Annals of Statistics*, **10**(4): 919-926.

[34] Pollard, D. (1990). Empirical Processes: Theory and Applications. *Nsf-Cbms Regional Conference Series in Probability and Statistics*, Vol.**2**.

[35] Rubin, D.B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, **9**(1): 130-134.

[36] Sarlós, T. (2006). Improved approximation algorithms for large matrices via random projections. *In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 143–152.

[37] Shao, J., Tu, D. (1995). The Jackknife and Bootstrap. *Springer Series in Statistics*, doi: 10.1007/978-1-4612-0795-5

[38] Singh, K., (1981). On the asymptotic accuracy of Eforn's bootstrap. *The Annals of Statistics*, **9**(6): 1187-1195.

[39] Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, **59**: 1-34.

[40] Teicher, H.(1974). On the law of the iterated logarithm. *Ann. Probability* **2**: 714–728.

[41] Vershynin, R. (2011). Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027v7*[math.PR].

[42] Wang, C., Chen, M.-H., Schifano, E., Wu, J. and Yan, J. (2015). A Survey of Statistical Methods and Computing for Big Data. arXiv:1502.07989

[43] Wang, H., Yang, M. and Stufken, J. (2017). Information-Based Optimal Subdata Selection for Big Data Linear Regression. To appear in *JASA*.

[44] White, Halbert (1980). A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica*, **48**(4): 817838. CiteSeerX 10.1.1.11.7646. doi:10.2307/1912934. JSTOR 1912934. MR 0575027.

[45] Zhu, R., Ma, P., Mahoney, M.W. and Yu, B. (2015). Optimal subsampling Approaches for Large Sample Linear Regression. *arXiv:1509.0511.v1* [stat.ME].

VITA

VITA

My name is Dali Zhou. I received my Bachelor's degree from Shandong Normal University in 2009. In 2016, I obtained my Master degree in Mathematics with concentration in Applied Statistics from the Department of Mathematical Sciences, IUPUI. Since then, I have continued my Ph.D. study in this department.