Distributed Supervised Statistical Learning

Amir Khalili Mahmoudabadi, Master of Science

Master of Science in Statistics

Submitted in partial fulfillment of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

Abstract

We live in the era of big data, nowadays, many companies face data of massive size that, in most cases, cannot be stored and processed on a single computer. Often such data has to be distributed over multiple computers which then makes the storage, pre-processing, and data analysis possible in practice. In the age of big data, distributed learning has gained popularity as a method to manage enormous datasets. In this thesis, we focus on distributed supervised statistical learning where sparse linear regression analysis is performed in a distributed framework. These methods are frequently applied in a variety of disciplines tackling large scale datasets analysis, including engineering, economics, and finance. In distributed learning, one key question is, for example, how to efficiently aggregate multiple estimators that are obtained based on data subsets stored on multiple computers. We investigate recent studies on distributed statistical inferences. There has been many efforts to propose efficient ways of aggregating local estimates, most popular methods are discussed in this thesis. Recently, an important question about the number of machines to deploy is addressed for several estimation methods, notable answers to the question are reviewed in this literature. We have considered a specific class of Liu-type shrinkage estimation methods for distributed statistical inference. We also conduct a Monte Carlo simulation study to assess performance of the Liu-type shrinkage estimation methods in a distributed computing environment.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Prof. Syed Ejaz Ahmed, for his unwavering support and guidance over these two years of my study and research at Brock University. I am truly grateful for his guidance and the valuable lessons I have learned under his supervision. I am immensely proud to have had the privilege of working with him.

My sincere thanks also extended to the external examiner Prof. Armin Hatefi, and to the committee members Prof. Tianyu Guan, and Prof. Pouria Ramazi for their valuable and constructive comments that helped me to improve this work.

I also would like to acknowledge my gratefulness to my mother and all my family members, and my especial thanks goes to my brother, Abbas, for all his support and encouragement.

Last but no least, I consider myself lucky to have a circle of my best friends over these two years of my study. Thank you, Reza, Azar, Nazanin and Kasra!

Contents

List of Figures

Chapter 1

Introduction

1.1 Linear regression analysis

Often, we are interested in studying (potential) relationship between variables. More specifically, we would like to study how one variable (or a vector of variables) depends on other variables. A statistical technique for examining the relationship between two continuous variables is linear regression analysis. It is a widely used method for modelling the relationship between a dependent variable (also known as the response or outcome variable) and one or more independent variables in data analysis (also called predictor or explanatory variables). Finding the optimum linear equation to describe the relationship between the variables is the aim of linear regression. Consider the following equation

$$y = \beta_0 + \beta_1 x + \varepsilon.$$

This is called simple linear regression or straight line regression, where y is dependent variable and x is independent variable. Term $\beta_0 + \beta_1 x$ is the systematic component and ε is the random component (error term). In this context, error does not mean mistake but is a statistical term representing random fluctuations, measurement errors, or the effect of factors outside of our control. Given a collection of observed data points, the linear regression procedure entails predicting the values of β_0 and β_1 that best match the data. To achieve this, the sum of the squared differences between the dependent variable's anticipated values and actual values must be minimized(Rencher and Schaalje (2008)).

Linear regression can be used for both simple and multiple regression analysis. Simple linear regression involves modelling the relationship between two variables, while multiple regression involves modelling the relationship between the dependent variable and two or more independent variables. As for multiple regression, a linear relationship between y_i's and x_{ij}'s for $i = 1, \ldots, N$ and $j = 1, \ldots, p$ has the matrix form of

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathbf{Y}_{N\times 1}$ is the response vector, $\mathbb{X}_{N\times p}$ is the design matrix (matrix of predictor variables), $\boldsymbol{\beta}_{p\times 1}$ is the vector of coefficients, and $\boldsymbol{\varepsilon}_{N\times 1}$ is the vector of errors.
Linear regression analysis is based on certain assumptions that must hold true for the results of the analysis to be valid and reliable. Violation of these assumptions may lead to incorrect conclusions or biased estimates. The key assumptions of linear regression analysis are:

1. **Linearity**: The relationship between the dependent variable and the independent variable(s) is linear. That is, the change in the dependent variable for a unit change in the independent variable(s) is constant.
2. **Independence**: The observations are independent of each other. That is, the value of one observation does not affect the value of another observation.
3. **Homoscedasticity**: The variance of the errors (residuals) is constant across all levels of the independent variable(s). That is, the spread of the residuals is the same at all values of the independent variable(s).
4. **Normality**: The residuals are normally distributed. That is, the errors follow a normal distribution.
5. **No or little multicollinearity**: There is no perfect (strong) linear relationship among the independent variables. That is, the independent variables are not highly correlated with each other.
6. **No influential outliers**: There are no extreme observations that have a large influence on the results of the analysis.

Violation of these assumptions can lead to biased estimates, incorrect confidence intervals, and incorrect hypothesis tests. Therefore, it is important to assess the assumptions of linear regression before conducting the analysis. Also to take appropriate measures to combat negative impacts caused by any violation of the assumptions.

Linear regression is a useful tool for making predictions and understanding the relationship between variables in a dataset. It has many applications in various fields such as economics, finance, marketing, and science.

1.2 Shrinkage estimation

Shrinkage methods (also known as regularization methods) are a class of linear regression estimation methods that start with the least-squares estimates and shrink the magnitude of some parameters towards zero. By doing so, we accept some bias in the estimation, but the hope is that we see a reduction in variance that outweighs the increase in bias and results in an overall reduction of mean-squared error. As an example, Least Absolute Shrinkage and Selection Operator (LASSO) shrinks the parameter estimates $\widehat{\beta}$ by imposing an l_1 penalty (absolute value) on their size. Shrinkage estimation is frequently used to model high-dimensional data or to increase robustness of the estimates in a variety of disciplines, including biology, economics, and finance. It is a potent strategy that can greatly increase the accuracy of parameter estimates and decrease the risk of overfitting. However, it requires careful tuning of the penalty parameter and selection of the appropriate method for the specific problem at hand. In section 1.6, we introduce a new class of shrinkage estimations proposed in Ahmed et al. (2023), Yüzbaşı et al. (2022) and Liu (2003).

1.3 Sparse linear regression

The idea of sparse linear regression analysis has been around for a long time. Sparsity has been a hot topic in statistics and machine learning since the LASSO was proposed in 1996. When there

are many predictors and many of them are unnecessary or redundant, sparse linear regression is a regression approach used to describe the relationship between a response variable and a set of predictor variables. The aim of sparse linear regression is to estimate the regression coefficients of the subset of relevant predictors while setting the regression coefficients of the irrelevant predictors to zero. To conduct this analysis, shrinkage estimation methods including Shrinkage methods such as LASSO, Ridge and Elastic net regression are used. Advantages to sparse coefficients vector are that for example, it prevents over-fitting, there are fewer parameters to estimate, and it makes interpreting the underlying phenomenon easier. Sparse linear regression has numerous applications in various fields, including genetics, finance, and image processing, where the number of predictors is often much larger than the sample size.

1.4 Analysis of big data

Big data often refers to data that is too big, chaotic, or dynamic for conventional databases or software packages to handle. Large, complex, and diversified data sets that are challenging to process and analyze using conventional data processing tools and techniques are referred to as big data. Using technology has resulted in the production of data; as a result, massively large amounts of data have been emerging. Many e-commerce businesses must analyse billions of transactions, social media support services must deal with datasets containing billions of user records, companies employing artificial intelligence (AI) to develop their products must deal with big data issues. To analyse such data, using one single computer is no longer efficient. Regarding the large number of observations to be handled, there are a number of challenges including limited processing power, and limited memory. Splitting big data into smaller subsets is a common technique used to process large volumes of data efficiently. This technique involves dividing the data into smaller partitions or subsets that can be processed on multiple machines or nodes simultaneously. Accordingly, distributed computing setup have become popular to tackle big data problems. Generally, for distributed learning in terms of statistical applications, there are primarily three directions: divide-and-conquer, online updating and subsampling.

Inspired by the idea of divide-and-conquer, parallel computing on a single machine and distributed computing have been executed. Both methods are dividing large tasks into smaller and more manageable tasks so that the process can be performed simultaneously on multiple CPUs or machines. The results are then aggregated as a final estimator by merging local estimates. In the parallel computing setting, the same memory is shared among the processors, and this causes a superefficient way to exchange the information, however the constraint on the memory limit is still effective. In distributed data processing (distributed computing), distinct machines are physically separated and connected to a central machine through a network, and there is no connection between local machines. Many studies of the issue have been done using the divide-and-conquer strategy. The review paper Gao et al. (2022), reviewed distributed statistical inferences and merging methods of finalizing the estimates gained by local computers. Furthermore, Zhao et al. (2016) considered a partially linear framework for modelling massive heterogeneous data. Battey et al. (2018) studied the topic on hypothesis testing and parameter estimation with a divide and conquer algorithm. Jordan et al. (2018) presented a communication-efficient surrogate likelihood framework for distributed statistical inference problems. The divide and conquer method for cubic-rate estimators under massive data framework was studied in Shi et al. (2018). Furthermore, Volgushev et al. (2019) proposed a two-step distributed method for quantile regression with data of massive

size.

As it was mentioned above, there is also online updating approach to conquer big-data problems. It focuses on big-data problems that observations are not available all at once. Online updating, which involves processing data in real-time or almost real-time as it becomes available, is a common method for handling big data. This method works especially well in real-time analytics, fraud detection, and recommendation systems, all of which require processing data of massive size rapidly and effectively. The observations come from a data stream in which the data is given in chunks sequentially. Some common techniques used for online updating in big data are: Stream Processing, Online Learning, Incremental Processing, and Approximate Computing. Different estimation strategies have been developed to be used in the framework of online updating approach. Schifano et al. (2016) developed some iterative estimating algorithms and statistical inference procedures for linear models and estimating equations with streaming data. Wang et al. (2018a) proposed an online updating method that could incorporate new variables for big data streams. Xue et al. (2020) proposed an online updating approach for testing the proportional hazards assumption with big survival data. Online updating, in general, is essential for many real-time and near real-time applications.

Another popular method is the subsampling approach, where the basic idea is to draw subsample for the purpose of statistical inferences. Ma et al. (2014) proposed an algorithmic leveraging-based sampling procedure. Wang et al. (2018b) and Wang (2019) developed some optimal subsampling methods for logistic regression with massive data. Wang et al. (2019) provided a novel information-based optimal subdata selection approach. Ai et al. (2021) studied the optimal subsampling algorithms for big data generalized linear models.Wang and Ma (2021) considered the optimal subsampling for quantile regression in big data. All these works have studied the case where the whole data is stored in one location. However, massive data are often distributed across multiple servers due to privacy, the storage burden, and computation abilities. For this problem, Zhang and Wang (2021) proposed a distributed subdata selection method for big data linear regression model. Particularly, they developed a two-step subsampling strategy with optimal subsampling probabilities and optimal allocation sizes. The subsample-based estimator effectively approximates the ordinary least squares estimator from the full data. Furthermore, the convergence rate and asymptotic normality of their proposed estimator were established.

1.5 Distributed learning

The study of distributed learning, which is a fast developing discipline, has the potential to completely alter how we handle and interpret massive datasets. Distributed learning has evolved into a crucial tool for data scientists and academics as a result of the emergence of big data and the rise in demand for sophisticated machine learning models. Leveraging the idea of divide and conquer, distributed learning is a powerful method for processing and analysing massive datasets. Distributed learning involves breaking a large and complex task down into smaller subtasks and distributing them over a number of computers or devices. As opposed to depending on a single, it is a powerful equipment to complete the task. After then, each device can complete the work that is assigned to it, and the final output can be created by combining the results.

Faster processing rates, increased accuracy, and the capacity to deal with bigger datasets than

would be possible on a single device are just a few advantages that distributed learning can provide. To guarantee that the results are precise and reliable, it also calls for careful control of data distribution and communication between equipment. Therefore, the idea of devide and conquer and distributed learning are two interconnected concepts that can be used to facilitate the process and analysis of enormous datasets. This combination offers a strong strategy which accelerates the workflows of data scientists and analysts, extracts insights from big data, and derives innovation in various industries.

In the context of statistical learning, distributed learning is referred by *distributed statistical inference*. A technique of doing statistical analysis on data that is split across several computing nodes or machines (local analyzer). Over the years, researchers have explored various aspects of this approach, including its theoretical foundations, algorithmic design, and applications in diverse domains. One of the most popular examples of distributed statistical learning is distributed sparse linear regression which refers to the process of performing the sparse linear regression analysis on local machines or nodes on their portion of data, and aggregate the results on a central machine.

1.5.1 Previous works

Distributed linear regression schemes refers to algorithms and methods that let many computing nodes work together to accomplish linear regression analysis after distributing datasets and even the data which is already distributed to be analyzed. These methods have been used in a variety of contexts, such as sensor networks, statistics, and machine learning. There has been a significant amount of research on distributed linear regression analysis under various settings, not necessarily sparsity, in recent years. As it is referred in Fonseca and Nadler (2023), see for example, Guestrin et al. (2004), Predd et al. (2006), Boyd et al. (2011), Duchi et al. (2014), McWilliams et al. (2014), Rosenblatt and Nadler (2016), Duchi et al. (2014), Chen et al. (2020), Dobriban and Sheng (2019), Zhu et al. (2021), and Dobriban and Sheng (2021).

For linear regression analysis under the sparsity assumption, Mateos et al. (2010), was among the first studies. They developed algorithms to perform LASSO estimation strategy when the training data is distributed across multiple agents and their communication to a central processing unit is prohibited due to some constraints such as communication costs or privacy. A motivating application mentioned in their paper is in the context of wireless communications: sensing cognitive radios collaborate to estimate the radio-frequency power spectrum density. The proposed algorithms aim to achieve a balance between complexity and convergence speed while minimizing inter-agent communication overhead. The algorithms use a separable form of the Lasso and the alternating-direction method of multipliers (ADMM) to iteratively minimize the objective function. The per agent estimate updates are given by simple soft-thresholding operations, and the inter-agent communication overhead is kept at an affordable level. The local estimates from each agent converge to the global lasso solution, which would be obtained if the entire data set were centrally available.

Numerical studies using both simulated and actual data showed that the proposed algorithms are successful. According to their work, the concepts can be expanded to match similar models, like the adaptive lasso, elastic net, fused lasso, and non-negative garrote, in a distributed manner. They considered a general setting where the local machines are not connected to a fusion center, however, they are connected and communicate with each other. Later efforts considered

the setting which we also consider in our work, where machines are connected only to a center machine and they are not allowed to communicate with each other but allowed to have one or more communication rounds with the center unit (known as iterative approach).

In the context of generalized sparse linear models, Chen and Xie (2014) proposed a divide (split)-and-conquer approach, where they distribute data over different machines to be analyzed by penalized estimation strategies. Each machine provides the fusion centre its sparse estimate, and the fusion centre estimates the support by voting on the indices of each machine's individual estimate. Finally, the central machine merges the estimates by weighted averaging as the final estimate. Under the sparsity assumption which tells each machine to use sparse linear regression, they used lasso strategy in each machine to compute local estimates. However, their method suffers from the well known bias of the lasso which is not reduced by averaging as an aggregation tool. Zhang and Zhang (2012); Javanmard and Montanari (2014), derived several debiasing techniques for the lasso and studied them theoretically. For distributed learning setup, debiased estimators have been used under various settings such as hypothesis testing, quantile regression and more, see for example Lee et al. (2017); Battey et al. (2018).

More specifically, under the high-dimensional setting, Lee et al. (2017) devised a distributed sparse regression method that is communication-efficient. In this method, "debiased" or "desparsified" lasso estimators are averaged. They demonstrate that, as long as the dataset is not split across too many machines, the method converges at the same rate as the lasso. In comparison to the lasso, the technique consistently predicts the support under weaker conditions. When the dataset is divided among samples, the authors suggest a new parallel and computationally effective algorithm to compute the approximate inverse covariance needed in the debiasing strategy. The approach is also expanded by the authors to include generalised linear models. Fonseca and Nadler (2023), studied distributed sparse linear regression under communication constraints, they considered two-round distributed schemes. After two rounds of communication, each machine computes a debiased lasso estimator and sends just a small subset of values to the fusion center. Their theoretical analysis demonstrates that the technique successfully recovers the exact support at low signal-to-noise ratios, where individual computers are unable to do so. Conducting simulations demonstrate that the method outperforms more communication-intensive techniques, sometimes even better. A related work to their study is Barghi et al. (2021).

According to their methodology, each machine computes a debiased lasso estimator but only sends the indices for which the absolute value of the estimate is greater than a certain threshold to the fusion centre. Any indices that were submitted by at least half of the machines, or indices that earned at least $K/2$ (K denotes the number of machines) votes, are included in the support set that the fusion centre estimates. In the context of feature selection, they derive bounds on the type-I and type-II errors of the estimated support set. The multiplicative constants are not specified by the authors, however, they do provide rates for these errors. However, Fonseca and Nadler (2023) showed that both theoretically and empirically, consistent support estimation is possible with a much lower voting threshold.

1.6 Liu-type shrinkage estimations in linear models

In a multiple linear regression model under sparsity assumption, it is usually assumed that the predictor variables are independent of each other. But, when the predictor variables are closed to be dependent, the multicollinearity could emerge. The Least Squares Estimator (LSE) is very sensitive if any of the considered assumptions in the model are violated. In this case, some biased estimators have been proposed. Proposed estimations to improve the least squares estimation are for example shrinkage estimation, principal components estimation, ridge estimation Hoerl and Kennard (1970), partial least squares estimation, Liu estimation Kejian (1993) and Liu-type estimation Liu (2003). To overcome multicollinearity, Yüzbaşı and Ejaz Ahmed (2016); Yüzbaşı et al. (2017) proposed the pretest and Stein-type ridge regression estimators for linear and partially linear models. Norouzirad and Arashi (2018) considered the preliminary test and Stein-rule Liu estimators for the ill-conditioned elliptical linear regression model. Alheety and Golam Kibria (2013) considered the modified Liu-type estimator for the linear regression model. Norouzirad and Arashi (2018) suggested a new rank-based Liu estimator as well as its shrinkage estimators. In this section we introduce methods proposed in Yüzbaşı et al. (2022) and Ahmed et al. (2023).

Consider the linear regression model (1.1) under the sparsity assumption. Under this assumption, vector of coefficients can be partitioned into two parts including a vector of main coefficients and a vector of insignificant coefficients. The method is estimating the main coefficients when insignificant coefficients are assumed to be close to zero. Accordingly, full-model and sub-model estimations can be considered; full-model estimation can incur high variability and sub-model estimation may cause underfitting with large bias. To combat these consequences, they consider pretest and shrinkage strategy to control magnitude of the bias. Consider a sparse linear model

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\mathbf{Y}_{N\times 1}$ is the vector of responses, $\mathbb{X}_{N\times p}$ is the design matrix or observation points, $\boldsymbol{\beta}_{p\times 1}$ is the vector of unknown regression coefficients, and $\boldsymbol{\varepsilon}_{N\times 1}$ is the vector of unobservable random errors. Additionally, assume that $\boldsymbol{\varepsilon}$ has a cumulative distribution function $\mathbf{F}(.)$; $\mathbf{E}(\boldsymbol{\varepsilon}) = 0$ and $Var(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}_N$, where σ^2 is finite and $\mathbf{I}_N$ is identity matrix of dimension $N \times N$. Furthermore, assume that the design matrix has rank p ($p \leq N$).

1.6.1 Estimation strategies

The ridge estimator firstly proposed by Hoerl and Kennard (1970), shrinks the parameter estimates $\widehat{\boldsymbol{\beta}}$ by imposing a penalty on their size. It includes a quadratic penalty term on the magnitude of $\widehat{\boldsymbol{\beta}}$ to the least-squares equation. Ridge Regression (RR) is a method for stabilizing regression estimates in the presence of extreme collinearity Frank and Friedman (1993). For model (1.1), the full-model ridge estimator is a solution to the following function

$$\arg\min_{\beta} (\mathbf{Y} - \mathbb{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbb{X}\boldsymbol{\beta}) + \lambda^R\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{1.2}$$

where the hyperparameter λ^R is the ridge parameter that controls the amount of shrinkage. The larger the value of λ^R the more shrinkage. Solving (1.2) yields

$$\widehat{\boldsymbol{\beta}}^{RFM} = (\mathbb{X}^T\mathbb{X} + \lambda^R\mathbf{I_p})^{-1}\mathbb{X}^T\mathbf{Y},$$

where $\widehat{\boldsymbol{\beta}}^{RFM}$ is called ridge full-model estimator. Obviously, if $\lambda^R = 0$, then $\widehat{\boldsymbol{\beta}}^{RFM}$ is the LSE estimator, and $\lambda^R = \infty$, then $\widehat{\boldsymbol{\beta}}^{RFM} = 0$. Kejian (1993) proposed a new class of estimators to combat multicollinearity. It was motivated by an interpretation of ridge estimate where they augmented $d\widehat{\boldsymbol{\beta}}^{LSE} = \boldsymbol{\beta} + \varepsilon^T$ to (1.1) and used least squares estimate to propose their estimate which has the form of

$$\widehat{\boldsymbol{\beta}}^{LFM} = (\mathbb{X}^T\mathbb{X} + \mathbf{I}_p)^{-1}(\mathbb{X}^T\mathbb{X} + d^L\mathbf{I}_p)\widehat{\boldsymbol{\beta}}_{LSE},$$

where $0 < d^L < 1$ and $\widehat{\boldsymbol{\beta}}^{LFM}$ is called Liu full-model estimator. The estimator is linear in d^L which is its advantage over $\widehat{\boldsymbol{\beta}}^{RFM}$. Their theory and simulation results showed that $\widehat{\boldsymbol{\beta}}^{LFM}$ has similar good properties as $\widehat{\boldsymbol{\beta}}^{RFM}$.

Now let $\mathbb{X} = (\mathbb{X}_1, \mathbb{X}_2)$, where $\mathbb{X}_1$ is an $N \times p_1$ sub-matrix including regressors of interest and $\mathbb{X}_2$ is an $N \times p_2$ sub-matrix including regressors which may or may not be relavent in the analysis of the main regressors. Similarly let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)^T$, where $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are column vectors of length p_1 and p_2 respectively, with $p_1 + p_2 = p$. Accordingly, the sub-model or restricted model is defined as

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \;\text{ subject to }\; \boldsymbol{\beta}_2 = 0,$$

subsequently, we have the following regression model as restricted or sub-model

$$\mathbf{Y} = \mathbb{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}. \tag{1.3}$$

The full-model or unrestricted ridge estimator of $\boldsymbol{\beta}_1$ is given by

$$\widehat{\boldsymbol{\beta}}_1^{RFM} = (\mathbb{X}_1^T\mathbb{M}_2^R\mathbb{X}_1 + \lambda^R\mathbf{I}_{p_1})^{-1}\mathbb{X}_1^T\mathbf{Y},$$

where $\mathbb{M}_2^R = \mathbf{I}_N - \mathbb{X}_2(\mathbb{X}_2^T\mathbb{X}_2 + \lambda^R\mathbf{I}_{p_2})^{-1}\mathbb{X}_2^T$ and λ^R is the ridge parameter for full-model estimator $\widehat{\boldsymbol{\beta}}_1^{RFM}$. The sub-model or restricted estimator of $\boldsymbol{\beta}_1$ is given by

$$\widehat{\boldsymbol{\beta}}_1^{RSM} = (\mathbb{X}_1^T\mathbb{X}_1 + \lambda_1^R\mathbf{I}_{p_1})^{-1}\mathbb{X}_1^T\mathbf{Y},$$

where λ_1^R is ridge parameter for sub-model estimator $\widehat{\boldsymbol{\beta}}_1^{RSM}$.

Similar to Ridge estimators, let us introduce the full-model or unrestricted Liu estimator that Liu proposed in Kejian (1993) for sub model. It is as follows

$$\widehat{\boldsymbol{\beta}}_1^{LFM} = (\mathbb{X}_1^T\mathbb{M}_2^L\mathbb{X}_1 + \mathbf{I}_{p_1})^{-1}(\mathbb{X}_1^T\mathbb{M}_2^L\mathbb{X}_1 + d^L\mathbf{I}_{p_1})\widehat{\boldsymbol{\beta}}_1^{LSE},$$

where $\mathbb{M}_2^L = \mathbf{I}_N - \mathbb{X}_2(\mathbb{X}_2^T\mathbb{X}_2 + \mathbf{I}_{p_2})^{-1}(\mathbb{X}_2^T\mathbb{X}_2 + d^L\mathbf{I}_{p_2})\mathbb{X}_2^T$ and $\widehat{\boldsymbol{\beta}}_1^{LSE} = (\mathbb{X}_1^T\mathbb{X}_1)^{-1}\mathbb{X}_1^T\mathbf{Y}$. The sub-model or restricted Liu estimator is defined as

$$\widehat{\boldsymbol{\beta}}_1^{LSM} = (\mathbb{X}_1^T\mathbb{X}_1 + \mathbf{I}_{p_1})^{-1}(\mathbb{X}_1^T\mathbb{X}_1 + d_1^L\mathbf{I}_{p_1})\widehat{\boldsymbol{\beta}}_1^{LSE},$$

where $0 < d_1^L < 1$.

In general terms, $\widehat{\boldsymbol{\beta}}_1^{LSM}$ performs better than $\widehat{\boldsymbol{\beta}}_1^{LFM}$ when $\boldsymbol{\beta}_2$ is close to zero. But, when $\boldsymbol{\beta}_2$ is not close to zero, $\widehat{\boldsymbol{\beta}}_1^{LSM}$ can be inefficient. However, $\widehat{\boldsymbol{\beta}}_1^{LFM}$ is consistent for $\boldsymbol{\beta}_2$ away from zero.

The idea of penalized estimation Chemometrics is a branch of chemistry that examines the use of statistical techniques in the study of chemical data. In addition to adopting numerous methods

from engineering and statistics literature, chemometrics itself has produced a number of unique data-analytical methods. Frank and Friedman (1993), examined principal components regression and partial least squares, two techniques that are frequently used in chemometrics for predictive modelling.

They aimed to comprehend the reasons behind their apparent successes, the circumstances in which they may be relied upon to perform well, and to contrast them with other statistical techniques designed for those circumstances. These techniques include ridge regression, variable subset selection, and ordinary least squares. Indeed, they introduced the idea of penalized estimation. In order to lessen the influence of the insignificant predictors on the result variable, they suggested a novel approach to regression analysis termed *bridge regression*, which entailed including a penalty term in the equation of standard regression.
This penalty term, known as the *bridge penalty*, controlled the degree of shrinkage of the regression coefficients towards zero, effectively limiting the impact of predictors that were not strongly related to the outcome variable. This method was eventually expanded to include other regression techniques, such ridge regression and lasso regression, and it has become a popular technique to used in a variety of domains, including statistics, machine learning, and data science, when dealing with high-dimensional data.

The notion of bridge regression they suggested is given in (1.4). For a given penalty function $\pi(.)$ and tuning parameter that controls the amount of shrinkage, λ, bridge estimators are estimated by minimizing the following penalized least square criterion,

$$\sum_{i=1}^{N}(y_i - x_i^T\beta)^2 + \lambda\pi(\beta), \tag{1.4}$$

where $\pi(\beta) = \sum_{j=1}^{p}|\beta_j|^\gamma$ and $\gamma > 0$. This penalty function bounds the L_γ norm of the parameters. The preliminary test, Stein-type and positive part Liu estimators in the linear models to minimize (1.4) were proposed and investigated in Ahmed et al. (2023) and Yüzbaşı et al. (2022).

A combination of $\widehat{\boldsymbol{\beta}}_1^{LFM}$ and $\widehat{\boldsymbol{\beta}}_1^{LSM}$ through an indicator function $\mathbf{I}(\mathscr{L}_N \leq c_{N,\alpha})$ is pretest estimator where $\mathscr{L}_N$ is an appropriate test statistic to test $H_0 : \boldsymbol{\beta}_2 = 0$ versus $H_A : \boldsymbol{\beta}_2 \neq 0$. Furthermore, $c_{N,\alpha}$ is an α-level critical value according to the distribution of $\mathscr{L}_N$. The test statistic is defined as follows:

$$\mathscr{L}_N = \frac{N}{\widehat{\sigma}^2}(\widehat{\boldsymbol{\beta}}_2^{LSE})^{-1}\mathbb{X}_2^T\mathbb{M}_1\mathbb{X}_2(\widehat{\boldsymbol{\beta}}_2^{LSE}),$$

where $\widehat{\sigma}^2 = \dfrac{1}{N-p}(\mathbf{Y}-\mathbb{X}\widehat{\boldsymbol{\beta}}^{LFM})^T(\mathbf{Y}-\mathbb{X}\widehat{\boldsymbol{\beta}}^{LFM})$, $\mathbb{M}_1 = \mathbb{I}_N - \mathbb{X}_1(\mathbb{X}_1^T\mathbb{X}_1)^{-1}\mathbb{X}_1^T$ and $\widehat{\boldsymbol{\beta}}_2^{LSE} = (\mathbb{X}_2^T\mathbb{M}_1\mathbb{X}_2)^{-1}\mathbb{X}_2^T\mathbb{M}_1\mathbf{Y}$.
Under H_0, the test statistic $\mathscr{L}_N$ follows chi-square distribution with p_2 degrees of freedom for large N values. The pretest Liu estimator is then defined by

$$\widehat{\boldsymbol{\beta}}_1^{LPT} = \widehat{\boldsymbol{\beta}}_1^{LFM} - (\widehat{\boldsymbol{\beta}}_1^{LFM} - \widehat{\boldsymbol{\beta}}_1^{LSM})\ \mathbf{I}(\mathscr{L}_N \leq c_{N,\alpha}).$$

The shrinkage or Stein-type Liu regression estimator $\widehat{\boldsymbol{\beta}}_1^{LS}$ of $\boldsymbol{\beta}_1$ is

$$\widehat{\boldsymbol{\beta}}_1^{LS} = \widehat{\boldsymbol{\beta}}_1^{LSM} + (\widehat{\boldsymbol{\beta}}_1^{LFM} - \widehat{\boldsymbol{\beta}}_1^{LSM})(1 - (p_2 - 2)\mathscr{L}_N^{-1}), p_2 \geq 3.$$

The estimator $\widehat{\boldsymbol{\beta}}_1^{LS}$ is general form of the Stein-rule family of estimators where shrinkage of the base estimator is towards the restricted estimator $\widehat{\boldsymbol{\beta}}_1^{LSM}$. The Shrinkage estimator is pulled towards the restricted estimator when the variance of the unrestricted estimator is large. Also, we can say that $\widehat{\boldsymbol{\beta}}_1^{LS}$ is the smooth version of $\widehat{\boldsymbol{\beta}}_1^{LPT}$. The positive part of shrinkage Liu regression estimator is given by

$$\widehat{\boldsymbol{\beta}}_1^{LPS} = \widehat{\boldsymbol{\beta}}_1^{LSM} + (\widehat{\boldsymbol{\beta}}_1^{LFM} - \widehat{\boldsymbol{\beta}}_1^{LSM})(1 - (p_2 - 2)\mathscr{L}_N^{-1})^+,$$

where $z^+ = max(0, z)$.
Furthermore, Yüzbaşı et al. (2022) and Ahmed et al. (2023) investigated the asymptotic properties of the estimators for $\gamma = 2$. The asymptotic bias of an estimator $\widehat{\boldsymbol{\beta}}_1^*$ is defined as

$$\mathcal{B}(\widehat{\boldsymbol{\beta}}_1^*) = \mathbb{E}\Big[\lim_{N\to\infty}\Big\{\sqrt{N}(\widehat{\boldsymbol{\beta}}_1^* - \boldsymbol{\beta}_1)\Big\}\Big],$$

where $\widehat{\boldsymbol{\beta}}_1^*$ is one of the suggested estimators. They showed that under some regularity conditions $\widehat{\boldsymbol{\beta}}^{LFM}$, is $\sqrt{N}$-consistent. Furthermore, let $\vartheta_1 = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LFM} - \boldsymbol{\beta}_1)$, $\vartheta_2 = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LSM} - \boldsymbol{\beta}_1)$, and $\vartheta_3 = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LFM} - \widehat{\boldsymbol{\beta}}_1^{LSM})$. Under similar regularity conditions, it was shown that $\begin{pmatrix}\vartheta_1\\ \vartheta_3\end{pmatrix}$ and $\begin{pmatrix}\vartheta_3\\ \vartheta_2\end{pmatrix}$ both asymptotically follow multivariate normal distribution and root N consistent.

The estimators introduced in this section, are to be used in the context of distributed learning. As we mentioned according to the results obtained in both Ahmed et al. (2023) and Yüzbaşı et al. (2022), their estimation strategies and their developed version of Liu-type estimators can be used for big data problems when the entire dataset is stored and processed by one single machine (centralized estimation). However, we aim to apply their estimation strategies for the case where the data has to be distributed across multiple machines. The hope is to see a comparable efficiency of their performance in the framework of distributed learning against that of an estimator based on all N samples.

Chapter 2

Distributed Statistical Inference

2.1 Distributed statistical inference setting

Consider a big-data setting where the sample size is large. Assume a large sample of size N denoted as $Z_i = (X_i^T, Y_i)^T \in \mathbb{R}^{p+1}, 1 \leq i \leq N$. Define $\{\mathbb{P}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ to be a family of statistical models parameterized by $\boldsymbol{\theta} \in \Theta$. Let the parameter space $\Theta \subset \mathbb{R}^p$ to be an open convex subset of Euclidian space. Additionally, assume that Z_i's have identical and independent $\mathbb{P}_{\boldsymbol{\theta}^*}$ distributions, where $\boldsymbol{\theta}^* = (\theta_1^*, \cdots, \theta_p^*)^T$ is the true parameter. Consider a standard architecture for a distributed computing system such that there are K local machines and a central machine denoted as $M_j, j = 1, ..., K$ and M_{center} respectively. Central machine is connected to every M_k and no connections are allowed among local machines. For a fixed N, split N sample units across K local machines randomly and evenly so that each local machine has $n = N/K$ observations. It should be noted that spliting is only done along observations not along variables. Write $\mathbb{S} = \{1, ..., N\}$ as the index set of whole sample. Then, let S_j denote the index set of local sample on M_j with $S_{j_1} \cap S_{j_2} = \emptyset$ for any $j_1 \neq j_2$. Let $\mathcal{L} : \Theta \times \mathbb{R}^{p+1} \mapsto \mathbb{R}$ be the loss function. Assume that the true parameter $\boldsymbol{\theta}^*$ minimizes the population risk $R(\boldsymbol{\theta}) = \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}; Z)]$, where $\mathbb{E}$ stands for expectation with respect to $\mathbb{P}_{\boldsymbol{\theta}^*}$. Define the local loss on the jth machine as

$$\mathcal{L}_j(\boldsymbol{\theta}) = n^{-1} \sum_{i \in S_k} \mathcal{L}(\boldsymbol{\theta}; Z_i).$$

Correspondingly, define the global loss function based on the whole sample as

$$\mathcal{L}(\boldsymbol{\theta}) = N^{-1} \sum_{i \in \mathbb{S}} \mathcal{L}(\boldsymbol{\theta}; Z_i) = K^{-1} \sum_{j=1}^{K} \mathcal{L}_j(\boldsymbol{\theta}),$$

whose minimizer is $\hat{\boldsymbol{\theta}} = arg\ min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta})$.

In most cases, the whole sample estimator $\hat{\boldsymbol{\theta}}$ is $\sqrt{N}$-consistent and asymptotically normal. Remember that an estimator is said to be consistent if it converges to the true value in probability. Additionally, once we know the estimator $\widehat{\boldsymbol{\theta}}$, it would be nice to know how quickly it converges to the true value $\boldsymbol{\theta}^*$, this is where $\sqrt{N}$-consistency comes to the business. The estimator is $\sqrt{N}$-consistent if $|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}| = O_p(1/\sqrt{N})$. Therefore, the efforts should be towards approaching this consistency when using distributed setting. As N is too large, it is very difficult (or even impossible in practice) to compute the whole sample estimator $\hat{\boldsymbol{\theta}}$ on one single machine. Hence, the distributed system

setting must be used. When the local estimators are all unbiased, it is clear that the simple average algorithm will yield an estimate that is as good as that of an estimator based on all N samples. Thus, the less bias in each local machine the less bias for the whole sample estimator. We have selected some estimation methods to be used in local machines (all local machines use the same method). Next we need to aggregate the estimates obtained from local computers to have a final estimate of the parameter of interest. All efforts have been made to reach an estimator as good as that of an estimator based on all N samples. Therefore, there are three major steps to take, splitting the whole dataset into smaller subsets, method of estimation that each single computer should use on its fraction of the data and an approach to aggregate local estimates on the central machine. Inspired by the idea of divide-and-conquer, various methods have been proposed that can be divided into two classes: One-shot and iterative approaches. One-shot method is to be discussed and is the method we use in the literature; iterative method is to be reviewed only.

2.2 Aggregation Strategies

As it was mentioned above, various methods have been proposed to solve the problem that can be roughly divided into one-shot and iterative classes. For both of them, different aggregator tools exist that have been introduced considering, for instance, MSE criterion and also based on the algorithm a practitioner chooses. Following sections will review the most popular merging strategies, however for our work, we focus on the simplest merging strategy, which is averaging the K local machine's estimates. The method is denoted as the Mixture Weight Method in Mcdonald et al. (2009). On the optimality of averaging in distributed statistical learning, a comprehensive study was accomplished by Rosenblatt and Nadler (2016), in section (2.4) we mention their results and use them to obtain our results.

2.2.1 One-shot approach

Basically, the idea of one-shot, also called simple average approach is to compute relevant statistics on each local machine, and send them to the central machine as the final stage. To do aggregation on M_{center}, the most popular and naive way is simple average. Local machines, M_j's, use their allocated sample and give us estimates, $\hat{\boldsymbol{\theta}}_j$'s. Then the estimates will be transferred to the center machine wherein $\widehat{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} = K^{-1} \sum_{j=1}^{K} \hat{\boldsymbol{\theta}}_j$ will be computed as a final answer (see figure (2.1)). Clearly, the one-shot setting is extremely communication-efficient. Because in this style each M_j communicates with M_{center} only once (there is only one single round of communication). Accordingly, the communication cost is of the order $O(Kp)$, where p is the dimension of $\hat{\boldsymbol{\theta}}$.

Properties of simple average as an algorithm to aggregate local results have been investigated by Zhang et al. (2012). We mention their results concisely in the subsection below.

2.2.2 Averaging methods

Duchi et al. (2014) introduced the simplest algorithm for distributed statistical inference called *average mixture* (AVGM) or simple average. In the AVGM algorithm, given a data set of size N and K local machines, first allocate a (distinct) data set of size $n = \frac{N}{K}$ randomly to each local machine, each machine gives us an estimate $\widehat{\boldsymbol{\theta}}_j$, and finally average all local estimates as a final

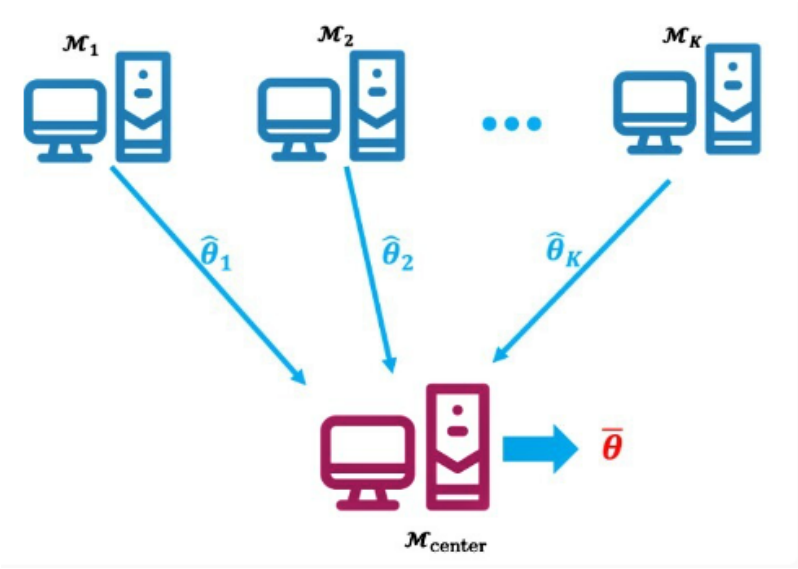

Fig. 2.1 *The illustration of the one-shot approach in distributed learning (taken from Gao et al. (2022)).*

answer (estimate). This method was used in Mcdonald et al. (2009), McDonald et al. (2010), and Zinkevich et al. (2010) as well. It is crucial to understand how effective the simple average estimator is. It is obvious that the simple average approach will produce an estimator that is as good as one based on all N samples when the local estimators are all unbiased. However, many estimators used in practice are biased. In Duchi et al. (2014) it was shown that, the mean squared error upper bound for simple average technique decays as $O(N^{-1} + n^{-2})$. If the number of machines is less than the size of sample per machine (i.e.,$K < n$), the simple average estimator reaches the best achievable rate which is possible in the case of *centralized estimation* or *gold standard* (i.e, where we use one single machine having access to the whole sample). One other contribution of their work was developing a novel extension of simple averaging, it is based on the idea of resampling proposed by Hastie et al. (2015); Hall (2013); Politis et al. (1999).

Duchi et al. (2014) referred to the algorithm as the *subsampled average mixture* (SAVGM). In the SAVGM algorithm similar to AVGM algorithm, it involves distributing the whole sample evenly and randomly across K local machines. However, instead of directly returning local estimates, each machine subsamples its own data to correct its estimate and provides a subsample-corrected estimate. The upper bound for its mean squared error was obtained by Duchi et al. (2014) and is of order $O(N^{-1} + n^{-3})$. When $K < n^2$, the method is first order equivalent with the centralized estimation and its second order term is smaller than the standard averaging approach. Furthermore, Duchi et al. (2014) studied sensitivity of the AVGM algorithm with respect to the number of local machines (K) and the SAVGM algorithm with respect to the amount of resampling. Their simulations showed that both methods have appropriate performance even when compared to the unattainable centralized approach that has access to the whole sample units. In Duchi et al. (2014), Corollary 2, it was shown that, under appropriate regularity conditions,

$$\mathbb{E}\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2^2 \leq \frac{C_1}{N} + \frac{C_2}{n^2} + O(\frac{1}{nN}) + O(\frac{1}{n^3}), \tag{2.1}$$

where C_1 and C_2 are positive constants. The boundary in (2.1) brings to our notice that, if n is large enough such that $n^{-2} = o(N^{-1})$, then the dominant term becomes C_1/N and is of the order

$O(N^{-1})$. This is the same as that of the whole sample estimator $\hat{\boldsymbol{\theta}}$. It implies that, to obtain the global convergence rate we should not divide the whole sample into too many parts. (Duchi et al. (2014))

Remark 2.2.1 In order to relax the constraint imposed on the number machines which is equivalent to not splitting the whole sample across too many machines, it was suggested one may increase the number of resampling rounds per machine. It is somehow performing like increasing the number of samples per machine. There is constraint on the number machines, because using too many machines causes smaller amount of sample units in each machine, thus we suggest increase in the number of machines but perform more resampling per machine to compensate lack of enough sample size in each machine. This is stated as a proposition below based on our understanding from the reviewing several literature working on the subject.

2.2.3 KL-divergence based combination method

Another technique to aggregate local estimates was proposed by Liu and Ihler in Liu and Ihler (2014) which is based on Kullback-Leibler divergence. In order to combine all the estimates as a final answer they used

$$\widehat{\boldsymbol{\theta}}_{KL} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}} \sum_{j=1}^{K} KL(p(x|\hat{\boldsymbol{\theta}}_j) \parallel p(x|\boldsymbol{\theta})),$$

where $p(x|\boldsymbol{\theta})$ is the probability density function of $\mathbb{P}_{\boldsymbol{\theta}}$ with respect to some proper measure μ, and KL-divergence is defined by $\mathrm{KL}(p(x)\|q(x)) = \int_{\mathcal{X}} p(x)\log\{p(x)/q(x)\}d\mu_x$. KL-divergence is basically a statistical distance that measures how two probability distributions are different from each other when one is the actual probability and the other one is an approximation of the actual probability. In the formula given above, $p(x)$ represents the actual and $q(x)$ the approximated probability distribution. In their paper, it was shown that $\widehat{\boldsymbol{\theta}}_{KL}$ is exactly the MLE estimate $\widehat{\boldsymbol{\theta}}$ for the whole sample.

There are many cases where the samples allocated to each machine are of poor quality which could affect local estimates and subsequently an inefficient final estimate. In this case, even the best possible estimators to be used per machine would fail. Therefore, a robust method to aggregate could help to tackle the problem. Minsker (2019) proposed a robust assembling method, where they compute the estimator by minimizing the following objective function

$$\widehat{\boldsymbol{\theta}}_{robust} = \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{j=1}^{K} \rho(|\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}|),$$

where the robust loss function $\rho(.)$ must satisfy some conditions mentioned in their paper. As an example in univariate case, where $p = 1$, when we have $\rho(x) = x$, the robust estimator $\widehat{\boldsymbol{\theta}}_{robust}$ is the median of $\widehat{\boldsymbol{\theta}}_j$'s. As we know, the median is more robust against outliers compare with the simple average. Under some regularity conditions, they showed that $\widehat{\boldsymbol{\theta}}_{robust}$ achieves the same convergence rate as the whole sample estimator provided $K \leq O(\sqrt{N})$.

2.2.4 Iterative approach

The one-shot approach benefits from low communication cost, simplicity and many other useful properties that are investigated in the literature have used the method so far. However, it has some drawbacks. First, the local machines need to have adequate amount of data, otherwise the local estimators would not be efficient, and the aggregated estimator cannot reach the convergence rate as the global estimator. This imposes limit on the number of machines that we can deploy, while we are interested in splitting the data across many machines (Duchi et al. (2014); Wang et al. (2017)). Second, the simple average to aggregate estimates is not suitable for nonlinear models Huang and Huo (2019); Jordan et al. (2018); Rosenblatt and Nadler (2016)). Third, when p diverges with N, some other failures could happen again, see Lee et al. (2017); Rosenblatt and Nadler (2016). It seems that some adjustment and rewinds of local estimators and the aggregator can come to the business as a remedy. As per iterative approach suggestions, a carefully designed iterative algorithm which uses a reasonable number of iterations could be useful for distributed systems. Inspired by the one-shot method in the M-estimator technique, Huang and Huo (2019) proposed a one-shot refinement of the simple averaging estimator (see figure(2.2)).

Recall the simple average estimator $\bar{\boldsymbol{\theta}}$ computed in the central machine for the one-shot approach, as we mentioned above, in order to improve the statistical efficiency of the final estimate, the modifications should be broadcast to each local machine. To do so, local gradient $\nabla\mathcal{L}_j(\bar{\boldsymbol{\theta}})$ and local Hessian $\nabla^2\mathcal{L}_j(\bar{\boldsymbol{\theta}})$ are computed on each local machine M_j. Next, they are transferred to M_{center} where they are aggregated by forming the whole sample gradient $\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}) = K^{-1}\sum_{j=1}^{K}\nabla\mathcal{L}_j(\bar{\boldsymbol{\theta}})$ and Hessian $\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}}) = K^{-1}\sum_{j=1}^{K}\nabla^2\mathcal{L}_j(\bar{\boldsymbol{\theta}})$. Then, the first round to update the aggregated estimate $\bar{\boldsymbol{\theta}}$ is as follows

$$\text{First iteration:} \quad \widehat{\boldsymbol{\theta}}^{(1)} = \bar{\boldsymbol{\theta}} - [\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}})]^{-1}\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}). \tag{2.2}$$

So far one more round of communication cost is added compared with the one-shot approach. Nevertheless, the statistical proficiency is well improved. Huang and Huo (2019) showed that

$$\mathbb{E}\|\widehat{\boldsymbol{\theta}}^{(1)} - \boldsymbol{\theta}^*\|_2^2 \leq \frac{C_1}{N} + O(\frac{1}{n^4}) + O(\frac{1}{N^2}),$$

where $C_1 > 0$ some constant. This is a lower upper bound for mean squared error than that in (2.1). It seems that even one additional round of communication has decreased the difference between the estimate and the true value of $\boldsymbol{\theta}$. To reach the convergence rate of the global estimator, size of the sample per machine need to satisfy $n^{-4} = o(N^{-1})$. The condition on the number local machines has become much milder. Therefore, we can roughly say that the first round of update in the iterative approach is equivalent to decreasing the number machines in the one-shot approach. Hence, the iterative method compared with the one-shot method, benefits from using more local process and more statistical efficiency simultaneously. Although, it has more communication cost which is getting more and more per each round of update. The idea is to allow the iteration (2.2) to be executed many times as long as the estimate is being improved. To generate the next step estimator, replace the updated estimator with its previous estimator, for instance, the second iteration

$$\text{Second iteration:} \quad \widehat{\boldsymbol{\theta}}^{(2)} = \widehat{\boldsymbol{\theta}}^{(1)} - [\nabla^2\mathcal{L}(\widehat{\boldsymbol{\theta}}^{(1)})]^{-1}\nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}^{(1)}).$$

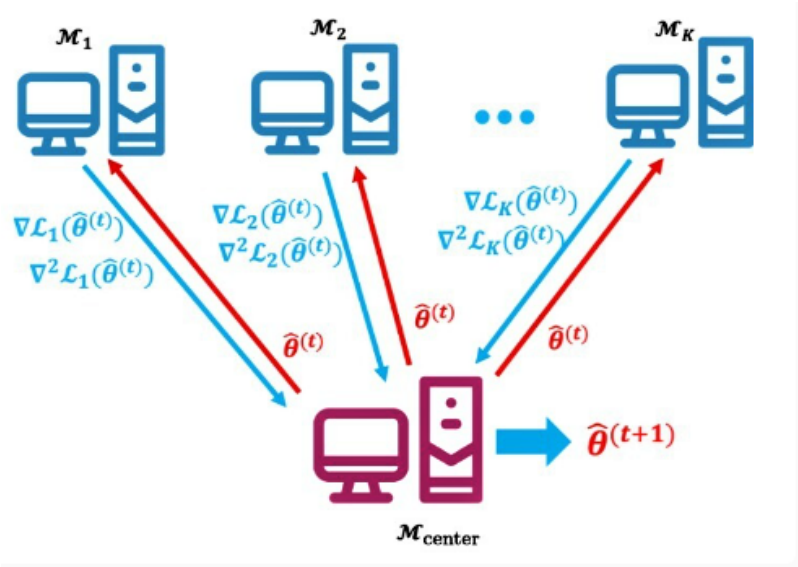

Fig. 2.2 *The illustration of the iterative approach in distributed learning (taken from Gao et al. (2022)).*

Similarly, for the tth iteration let $\widehat{\boldsymbol{\theta}}^{(t)}$ be the estimator such that

$$(t+1)\text{th iteration:} \quad \widehat{\boldsymbol{\theta}}^{(t+1)} = \widehat{\boldsymbol{\theta}}^{(t)} - [\nabla^2\mathcal{L}(\widehat{\boldsymbol{\theta}}^{(t)})]^{-1}\nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}^{(t)}).$$

As it is obvious, this requires a large number of Hessian matrices to be computed and transferred. When the dimension of parameter p is high, there will be a large volume of computation which is infeasible or at least difficult in practice to be executed. Cost of the process will be of the order $O(Kp^2)$ that is significantly large when the p is relatively large. To fix the posed problem, Shamir et al. (2014) proposed an approximate Newton method, it applies the Newton-type iteration distributedly to the distributed system but without transferring the Hessian matrices. According to this strategy, an approximate likelihood approach was developed by Jordan et al. (2018), the idea was to update Hesian matrix on only one machine (e.g., M_{center}). Following their idea, the iteration can be modified to be

$$\widehat{\boldsymbol{\theta}}^{(t+1)} = \widehat{\boldsymbol{\theta}}^{(t)} - [\nabla^2\mathcal{L}_{center}(\widehat{\boldsymbol{\theta}}^{(t)})]^{-1}\nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}^{(t)}),$$

where $\nabla^2\mathcal{L}_{center}$ is the Hessian matrix computed on the central machine. The strategy involves less communication cost as it is not transferring the Hessian matrices anymore. In Jordan et al. (2018), it was shown that under some conditions

$$\|\widehat{\boldsymbol{\theta}}^{(t+1)} - \widehat{\boldsymbol{\theta}}\|_2 \leq \frac{C_1}{\sqrt{n}}\|\widehat{\boldsymbol{\theta}}^{(t)} - \widehat{\boldsymbol{\theta}}\|_2, \quad \text{for } t \geq 0, \tag{2.3}$$

where $C_1 > 0$ is constant and $\widehat{\boldsymbol{\theta}}$ is the whole sample estimator. Obviously, there must be a limit on the number iterations. From the linear convergence formula (2.3), we can see that it requires $[\log K/\log n]$ iterations to achieve the $\sqrt{N}$-consistency as the whole sample estimator $\widehat{\theta}$, provided $\widehat{\theta}^{(0)}$ is $\sqrt{n}$-consistent. If $n = K = \sqrt{N}$, one iteration suffices to attain the optimal convergence rate Gao et al. (2022).

However, the effective selection of the machine on which the Hessian matrix is to be updated is crucial for the procedure to work as intended Fan et al. (2021). In Fan et al. (2021), they added an extra regularization term to the likelihood used in Jordan et al. (2018) to improve the solution.

2.2.5 Popular shrinkage methods

Shrinkage methods are used for sparse estimation. For a high-dimensional problem, especially when the dimension of $\boldsymbol{\theta}^*$ is larger than the sample size N, it is difficult to estimate $\boldsymbol{\theta}^*$ without any additional assumptions Hastie et al. (2015). A popular constraint to be considered is sparsity, where $\boldsymbol{\theta}^*$ is partitioned into two parts: zero and non-zero entries. The index of non-zero entries is called the support of $\boldsymbol{\theta}^*$, that is

$$supp(\boldsymbol{\theta}^*) = \mathcal{A}^* = \{1 \leq j \leq p : \boldsymbol{\theta}_j^* \neq 0\}.$$

The specific shrinkage regression problem to be solved can be given by

$$\min_{\boldsymbol{\theta}\in\Theta}\{\mathcal{L}(\boldsymbol{\theta}) + \sum_{j=1}^{p} \rho_\lambda(|\boldsymbol{\theta}_j|)\},$$

where $\rho_\lambda(.)$ is a penalty function with a regularization hyperparameter $\lambda > 0$. Popular choices as a penalty function are LASSO (least absolute shrinkage and selection operator) proposed in Tibshirani (1996), SCAD (smoothly clipped absolute deviation) proposed in Fan et al. (2021) and others discussed in Zhang and Zhang (2012). In this literature in chapter two we use a new class of shrinkage estimators introduced by Yüzbaşı et al. (2022).

Consider the LASSO estimator for the linear regression problem. The whole sample estimator is an argument in the parameter space which minimizes the function given below

$$\widehat{\boldsymbol{\theta}}_\lambda = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\frac{1}{N}\|\mathbf{Y} - \mathbb{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1\},$$

where $\mathbf{Y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ is the vector of response and $\mathbb{X} \in \mathbb{R}^{N\times p}$ is the design matrix, and $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^p |\theta_j|$ is the l_1-norm of $\boldsymbol{\theta}$. For large coefficients, it has been observed that the LASSO procedure may produce biased estimators. On the other hand, the simple average techniques to aggregate local estimators, often, do not perform efficiently to eliminate and alleviate the systematic bias. A debiasing technique was proposed by Javanmard and Montanari (2014) which is given by

$$\widehat{\boldsymbol{\theta}}_\lambda^{(d)} = \widehat{\boldsymbol{\theta}}_\lambda + \frac{1}{N}\mathbb{M}\mathbb{X}^T(\mathbf{Y} - \mathbb{X}\widehat{\boldsymbol{\theta}}_\lambda), \tag{2.4}$$

where $\mathbb{M} \in \mathbb{R}^{p\times p}$ is an approximation to the inverse of $\widehat{\Sigma} = \mathbb{X}^T\mathbb{X}/N$. When $\widehat{\Sigma}$ is invertible (e.g., $N \gg p$), setting $\mathbb{M} = (\widehat{\Sigma})^{-1}$ gives $\widehat{\boldsymbol{\theta}}_\lambda^{(d)} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{Y}$ which is the ordinary least squares estimator and unbiased. Therefore, the debiasing technique (2.4), compensates the bias caused by l_1 regularization in some sense. For distributed setting, the debiasing procedure could be used on each local machine, and averaging estimator can be constructed on the central machine M_{center}. The procedure was used by Lee et al. (2017), they developed a one-shot type estimator for the LASSO problem. Each local machine gives the estimator $\widehat{\boldsymbol{\theta}}_{k,\lambda}^{(d)}$ then an averaging estimator on the central machine is given as $\bar{\boldsymbol{\theta}}_\lambda = \sum_{k=1}^{K} \widehat{\boldsymbol{\theta}}_{k,\lambda}^{(d)}$.

Shrinkage methods are considered under the sparsity assumption, however, averaging can degrade the level of sparsity dramatically. To handle this problem, a hard threshold step often can come as a remedy. Additionally, the debiasing step mentioned above, is computationally expensive. To alleviate the computational cost of this step, Lee et al. (2017) proposed an improved algorithm.

They showed that, under some conditions, the resulting estimator reaches the same convergence rate as the whole sample estimator. As mentioned before, the one-shot approach needs a stringent constraint on per-machine sample size and accordingly on the number of processors due to the limited communication, which is undesirable. For sparse linear model, averaging estimator requires $n \geq O(Ks^2 \log p)$ to approach the efficiency of the whole sample estimator (Lee et al. (2017)), where $s = |\mathcal{A}^*|$ is the number of non-zero entries of $\boldsymbol{\theta}^*$.

For this problem, Wang et al. (2017) and Jordan et al. (2018) proposed communication-efficient iterative algorithm which constructs a regularized likelihood by using local Hessian matrix. Wang et al. (2017) showed that even if $n \geq O(Ks^2 \log p)$, one iteration is enough to have the convergence rate of $\widehat{\boldsymbol{\theta}}^{(1)}$ close to the global convergence rate. However, if multi-round communication is allowed, $\widehat{\boldsymbol{\theta}}^{(t+1)}$ is equivalent with the whole sample estimator as long as $n \geq O(s^2 \log p)$ and $t > O(\log p)$, under some conditions. Hence, their method can relax the constraint on the number machines while it tells us that number iterations has a lower bound, that is, it must be greater than the order $O(\log p)$. One may start from a number pretty close to the lower bound they provided so that the costs of communication and computation do not grow.

2.3 Distributed Liu-type shrinkage estimations for sparse linear models

In previous sections, the methodology for distributed statistical inference was investigated, and aggregating strategies were reviewed. In section (2.2.5), the distributed setting was discussed for debiased shrinkage estimation strategies. Shrinkage methods are a family of statistical techniques used to reduce the complexity of models and prevent overfitting. In the context of distributed learning, shrinkage methods can be used to improve the performance of distributed sparse linear regression algorithms. Yüzbaşı et al. (2017), showed that their estimation strategies have comparable performance with other efficient methods, however, it was studied for centralized estimation. Furthermore, Pretest and other discussed shrinkage estimators were proposed to combat multicollinearity. Negative effects of multicollinearity are magnified at smaller sample sizes. On the other side, we split the data across multiple machines, thus, the sample size in each machine would obviously be small. Subsequently, local machines suffer from the magnified multicollinearity negative effects due to smaller sample sizes. The ability of these estimators to handle multicollinearity motivated us to use them on each local machine. Now, let us consider their methods of shrinkage estimations introduced in section (1.6) to be used in a distributed framework.

2.3.1 Problem setup

Consider a linear regression model

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.5}$$

where $\mathbf{Y}$ is the response vector of length N, $\mathbb{X}$ is the design matrix of dimension $N \times p$ which includes observation points, $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_p)^T$ is the vector of unknown regression coefficients, and $\boldsymbol{\varepsilon}$ is the vector of unobservable random errors. We also assume that the design matrix $\mathbb{X}$ has rank p ($p \leq N$) and $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_N)^T$ has a cumulative distribution function $F(.)$; $E(\boldsymbol{\varepsilon}) = 0$ and $Var(\boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}_N$, where σ^2 is finite and $\mathbf{I}$ is an identity matrix of dimension $N \times N$. Furthermore, consider the sparsity assumption for the linear regression model (2.5). Under this assumption, the

vector of coefficients $\boldsymbol{\beta}$ can be partitioned as $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)^T$ where $\boldsymbol{\beta}_1$ is the vector of coefficients for main effects, and $\boldsymbol{\beta}_2$ is the vector for nuisance effects or insignificant coefficients, i.e, the regression coefficients are p_1-sparse. Our interest is to divide the dataset into smaller chunks, and apply the shrinkage methods discussed in section (1.6) on each portion of sample. For a fixed N, split N sample units across K local machines randomly and evenly so that each local machine has $n = N/K$ observations. It should be noted that splitting is only done along observations not along variables.

Hence, given the number of machines, K, the design matrix $\mathbb{X}$ is split into K submatrices randomly and evenly. Therefore, in each local machine M_j, we have

$$\boldsymbol{Y}_j = \mathbb{X}_j\boldsymbol{\beta} + \boldsymbol{\varepsilon}_j, \quad j = 1, \ldots, K, \tag{2.6}$$

where $\mathbb{X}_j$ is an $n \times p$ local design matrix, $\boldsymbol{Y}_j$ is the vector of local responses with length n, and the vector of local random error $\boldsymbol{\varepsilon}_j$ with length n. Obviously, model (2.6) is still under the sparsity and all other assumptions mentioned above.

2.3.2 Estimation strategies

Now, for each local machine M_j $(j = 1, \ldots, K)$, let us define the estimators using distributed setting notations. The local ridge full model estimator is given by

$$\widehat{\boldsymbol{\beta}}_j^{RFM} = (\mathbb{X}_j^T\mathbb{X}_j + \lambda_j^R\mathbf{I}_p)^{-1}\mathbb{X}_j\mathbf{Y}_j, \tag{2.7}$$

where $0 < \lambda_j^R < 1$ is the ridge parameter in machine j for the dataset $\mathbb{X}_j$,$\mathbf{Y}_j$. Liu full model biased estimator (LFM), is defined as

$$\widehat{\boldsymbol{\beta}}_j^{LFM} = (\mathbb{X}_j^T\mathbb{X}_j + \mathbf{I}_p)^{-1}(\mathbb{X}_j^T\mathbb{X}_j + d_j^L\mathbf{I}_p)\widehat{\boldsymbol{\beta}}_j^{LSE}, \tag{2.8}$$

where $0 < d_j^L < 1$ is the local Liu parameter and $\widehat{\boldsymbol{\beta}}_j^{LSE} = (\mathbb{X}_j^T\mathbb{X}_j)^{-1}\mathbb{X}_j^T\mathbf{Y}_j$.

For local machines, under the sparsity assumption, let us partition the design matrix in each machine as $\mathbb{X}_j = (\mathbb{X}_{1j}, \mathbb{X}_{2j})$, where $\mathbb{X}_{1j}$ is an $n \times p_1$ sub-matrix containing the regressors of interest and $\mathbb{X}_{2j}$ is an $n \times p_2$ sub-matrix that may or may not be relevant in the analysis of the main regressors. Similarly, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)^T$ be the vector of parameters, where $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ have dimensions p_1 and p_2, respectively, with $p_1 + p_2 = p$.

The sub-model or restricted model in each local machine M_j is defined as:

$$\mathbf{Y}_j = \mathbb{X}_j\boldsymbol{\beta} + \boldsymbol{\varepsilon}_j \ \text{ subject to } \ \boldsymbol{\beta}_2 = 0,$$

then we have the following restricted linear regression model in machine M_j,

$$\mathbf{Y}_j = \mathbb{X}_{1j}\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}_j. \tag{2.9}$$

$\widehat{\boldsymbol{\beta}}_{1j}^{RFM}$ denoted as the full model or unrestricted ridge estimator of $\boldsymbol{\beta}_1$ is given by

$$\widehat{\boldsymbol{\beta}}_{1j}^{RFM} = (\mathbb{X}_{1j}^T\mathbb{M}_{2j}^R\mathbb{X}_{1j} + \lambda_j^R\mathbf{I}_{p_1})^{-1}\mathbb{X}_{1j}^T\mathbb{M}_{2j}^R\mathbb{Y}_j,$$

where $\mathbb{M}_{2j}^R = \mathbf{I}_n - \mathbb{X}_{2j}(\mathbb{X}_{2j}^T\mathbb{X}_{2j} + \lambda_j^R\mathbf{I}_{p_2})^{-1}\mathbb{X}_{2j}^T$. For model (2.9), the sub-model or restricted estimator $\widehat{\boldsymbol{\beta}}_{1j}^{RSM}$ of $\boldsymbol{\beta}_1$ has the form

$$\widehat{\boldsymbol{\beta}}_{1j}^{RSM} = (\mathbb{X}_{1j}^T\mathbb{X}_{1j} + \lambda_{1j}^R\mathbf{I}_{p_1})^{-1}\mathbb{X}_{1j}^T\mathbf{Y}_j,$$

where λ_{1j}^R is ridge parameter for the sub-model estimator $\widehat{\boldsymbol{\beta}}_{j1}^{RSM}$ in machine j.
Introduced full model or unrestricted Liu estimator $\widehat{\boldsymbol{\beta}}_{1j}^{LFM}$ to be used by each machine is given by

$$\widehat{\boldsymbol{\beta}}_{1j}^{LFM} = (\mathbb{X}_{1j}^T\mathbb{M}_{2j}^L\mathbb{X}_{1j} + \mathbf{I}_{p_1})^{-1}(\mathbb{X}_{1j}^T\mathbb{M}_{2j}^L\mathbb{X}_{1j} + d_j^L\mathbf{I}_{p_1})\widehat{\boldsymbol{\beta}}_{1j}^{LSE},$$

where $\mathbb{M}_{2j}^L = \mathbf{I}_n - \mathbb{X}_{2j}(\mathbb{X}_{2j}^T\mathbb{X}_{2j} + \mathbf{I}_{p_2})^{-1}(\mathbb{X}_{2j}^T\mathbb{X}_{2j} + d_j^L\mathbf{I}_{p_2})\mathbb{X}_{2j}^T$ and $\widehat{\boldsymbol{\beta}}_{1j}^{LSE} = (\mathbb{X}_{1j}^T\mathbb{X}_{1j})^{-1}\mathbb{X}_{1j}^T\mathbf{Y}$.
The sub-model Liu estimator will be

$$\widehat{\boldsymbol{\beta}}_{1j}^{LSM} = (\mathbb{X}_{1j}^T\mathbb{X}_{1j} + \mathbf{I}_{p_1})^{-1}(\mathbb{X}_{1j}^T\mathbb{X}_{1j} + d_{1j}^L\mathbf{I}_{p_1})\widehat{\boldsymbol{\beta}}_{1j}^{LSE},$$

where $0 < d_{1j}^L < 1$ and $\widehat{\boldsymbol{\beta}}_{1j}^{LSE} = (\mathbb{X}_{1j}^T\mathbb{X}_{1j})^{-1}\mathbb{X}_{1j}^T\mathbf{Y}$.

Similar to the centralized estimation, as it was mentioned before, when $\boldsymbol{\beta}_2$ is close to zero, $\widehat{\boldsymbol{\beta}}_{1j}^{LSM}$ performs better than $\widehat{\boldsymbol{\beta}}_{1j}^{LFM}$. However, for $\boldsymbol{\beta}_2$ away from zero, $\widehat{\boldsymbol{\beta}}_{1j}^{LSM}$ can be inefficient. But, $\widehat{\boldsymbol{\beta}}_{1j}^{LFM}$ is consistent for departure of $\boldsymbol{\beta}_2$ from zero Yüzbaşı et al. (2017).

Pretest and Shrinkage Liu estimation

The pretest is a combination of $\widehat{\boldsymbol{\beta}}_{1j}^{LFM}$ and $\widehat{\boldsymbol{\beta}}_{1j}^{LSM}$ through an indicator function $\mathbf{I}(\mathscr{L}_j \leq c_{j,\alpha})$, where $\mathscr{L}_j$ is an appropriate test statistic to test $H_0 : \boldsymbol{\beta}_2 = 0$ versus $H_A : \boldsymbol{\beta}_2 \neq 0$. Moreover, $c_{j,\alpha}$ is an α-level critical value using the distribution of $\mathscr{L}_j$. The test statistic is defined as follows:

$$\mathscr{L}_j = \frac{n}{\widehat{\sigma}_j^2}(\widehat{\boldsymbol{\beta}}_{2j}^{LSE})^T\mathbb{X}_{2j}^T\mathbb{M}_{1j}\mathbb{X}_{2j}(\widehat{\boldsymbol{\beta}}_{2j}^{LSE}),$$

where $\widehat{\sigma}_j^2 = \frac{1}{n-p}(\mathbf{Y}_j - \mathbb{X}_j\widehat{\boldsymbol{\beta}}_j^{LFM})^T(\mathbf{Y}_j - \mathbb{X}_j\widehat{\boldsymbol{\beta}}_j^{LFM})$ is consistent estimator of σ^2, $\mathbb{M}_{1j} = \mathbf{I}_{nj} - \mathbb{X}_{1j}(\mathbb{X}_{1j}^T\mathbb{X}_{1j})^{-1}\mathbb{X}_{1j}^T$ and $\widehat{\boldsymbol{\beta}}_{2j}^{LSE} = (\mathbb{X}_{2j}^T\mathbb{M}_{1j}\mathbb{X}_{2j})^{-1}\mathbb{X}_{2j}^T\mathbb{M}_{1j}\mathbf{Y}_j$. Under H_0, the test statistic $\mathscr{L}_j$ follows chi-squared distribution with p_2 degrees of freedom for large n values. Now, the pretest Liu regression estimator $\widehat{\boldsymbol{\beta}}_{1j}^{LPT}$ of $\boldsymbol{\beta}_1$ is defined by

$$\widehat{\boldsymbol{\beta}}_{1j}^{LPT} = \widehat{\boldsymbol{\beta}}_{1j}^{LFM} - (\widehat{\boldsymbol{\beta}}_{1j}^{LFM} - \widehat{\boldsymbol{\beta}}_{1j}^{LSM})\mathbf{I}(\mathscr{L}_j \leq c_{j,\alpha}),$$

where $c_{j,\alpha}$ is an α-level critical value. The Shrinkage or Stein-type Liu regression estimator $\widehat{\boldsymbol{\beta}}_{1j}^{LS}$ of $\boldsymbol{\beta}_1$ is defined by

$$\widehat{\boldsymbol{\beta}}_{1j}^{LS} = \widehat{\boldsymbol{\beta}}_{1j}^{LSM} + \left(\widehat{\boldsymbol{\beta}}_{1j}^{LFM} - \widehat{\boldsymbol{\beta}}_{1j}^{LSM}\right)\left(1 - (p_2 - 2)\mathscr{L}_j^{-1}\right), \quad p_2 \geq 3.$$

The positive part of the shrinkage Liu regression estimator of $\boldsymbol{\beta}_1$ denoted by $\widehat{\boldsymbol{\beta}}_{1j}^{LPS}$ is given by

$$\widehat{\boldsymbol{\beta}}_{1j}^{LPS} = \widehat{\boldsymbol{\beta}}_{1j}^{LSM} + \left(\widehat{\boldsymbol{\beta}}_{1j}^{LFM} - \widehat{\boldsymbol{\beta}}_{1j}^{LSM}\right)\left(1 - (p_2 - 2)\mathscr{L}_j^{-1}\right)^+,$$

where $z^+ = max(0, z)$.

2.3.3 Aggregation

So far, we have defined the local shrinkage methods to perform local estimation in each local machine, however, to obtain the final answer (estimate) we need to aggregate them in the center machine. As it was mentioned before, simple average is the aggregation method which is used in this literature. The aggregated estimator (the whole sample estimator) for introduced estimation strategies is given by:

$$\widehat{\boldsymbol{\beta}} = \frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{\blacktriangle}, \tag{2.10}$$

where $\widehat{\boldsymbol{\beta}}_j^{\blacktriangle}$ is one of the discussed shrinkage estimators: RFM, LFM, RFM1, RSM1, LFM1, LSM1, LPT1, LS1, LPS1.

Remark 2.3.1 It is required to understand if the setting under which the averaging technique is efficient, matches the setting for the estimators we intend to apply in each machine. Furthermore, it assists us to obtain a boundary for the number of machines to deploy (according to the estimation strategies).

As we consider averaging technique as the aggregation strategy, it is required to provide more detail on the optimality of averaging. In the next section, simple average aggregation technique and its properties are reviewed.

2.4 On the optimality of averaging

Content of this section is derived from the work of Rosenblatt and Nadler (2016) on the optimality of averaging.

2.4.1 Introduction

The focus of Rosenblatt and Nadler (2016) is on the *statistical properties* of the split-and-merge approach, under the assumption that the observations are split uniformly at random among the K machines. In the context of simple average method which is the approach we follow in our work, they addressed following questions: (i) What is the estimation error of simple averaging as compared with a centralized solution? (ii) What is its distribution? (iii) Under which criteria, if any, is averaging optimal? and (iv) How many machines to deploy? They also referred to some other works such as McDonald et al. (2010)(Theorem 3) that were among the first to study some of these issues for multinomial regression, deriving finite sample bounds on the expected error of the averaged estimator. Zinkevich et al. (2010) compared the statistical properties of the averaged estimator to the centralized estimation for more general learning tasks, assuming each machine estimates the model parameters by stochastic gradient descent. More recently, under appropriate conditions and for a large class of loss functions, Duchi et al. (2014) derived bounds for the leading order term in the mean squared error (MSE) of the averaged estimator and provided the rates of higher-order terms. They further proposed several improvements to the simple averaging strategy that reduce the second-order term in the MSE, and reduce the machine-wise run time via modified optimization algorithms.

Rosenblatt and Nadler (2016) extended and generalized these previous studies in several aspects. Studying the statistical properties of the averaged (merged) estimator, when the number of parameters p is fixed under some conditions. Using the theory of M-estimators Rieder (2012) and Van der Vaart (2000) provided not only asymptotic bounds on the MSE, but rather an asymptotic expansion of the error itself. This allows them to derive the exact constants in the MSE expansion, and prove that as $n \to \infty$, the MSE of the averaging strategy in fact equals to that of the centralized solution. In other words, when the number of machines K and their available memory are such that in each machine there are many observations per parameter ($n \gg p$), then averaging machine-wise estimates is as accurate as the centralized solution. Furthermore, if the centralized estimator possesses first-order statistical properties such as efficiency and robustness, then so will the estimator obtained from the distributed setting (averaged estimator). It is remarkable that, the first-order equivalence between averaged and centralized estimators seems to be a good deal as we are deploying more than one machine to decrease the run-time without loss of accuracy. However, distributed estimation via split-and-average does incur an accuracy loss captured in the higher-order error terms Rosenblatt and Nadler (2016). In the following sections we will go through more details on these statistical properties towards our goal which is applying the new class of shrinkage estimations proposed in Yüzbaşı et al. (2022).

Consider the setting in section (2.1), where we let $\boldsymbol{\theta}^* \in \Theta$ be the minimizer of the population risk defined as follows

$$R(\boldsymbol{\theta}) := \mathbb{E}_Z[\mathcal{L}(\boldsymbol{\theta}; Z)] = \int \mathcal{L}(\boldsymbol{\theta}; Z) dP_Z(z). \tag{2.11}$$

Similar to previous setting, assume that $\boldsymbol{\theta}^*$ exists in Θ and is unique. Given N i.i.d. samples $\{z_i\}_{i=1}^N$ of the random variable Z, a standard approach, known as M-estimation or empirical risk minimization (ERM), is to calculate the estimator $\hat{\boldsymbol{\theta}}_N \in \Theta$ that minimizes the empirical risk

$$\hat{R}_N(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(z_i, \boldsymbol{\theta}). \tag{2.12}$$

2.4.2 Fixed-p setting

Similar to Rosenblatt and Nadler (2016), in the distributed learning setup, let us first consider the regime where p is fixed. First, they consider the error of the split-and-average estimator $\bar{\boldsymbol{\theta}}$. In this regime, the model dimension p and number of machines K are both fixed. Instead of focusing on the MSE, they derive an exact asymptotic representation of the first two terms in the error $\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$ itself.

First-order statistical properties of averaging

Starting by analysis of the exact asymptotic expression for the dominant error term, we consider the following standard assumptions made in Rosenblatt and Nadler (2016). In this section we follow their notations that $\boldsymbol{\theta}_n$ denotes local estimators.

Assumption Set 1

(A1) Machine-wise estimators, $\hat{\boldsymbol{\theta}}_n$'s, are consistent: $\hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}^* + o_P(1)$,

(A2) $R(\boldsymbol{\theta})$ admits a second-order Taylor expansion at $\boldsymbol{\theta}^*$ with non-singular Hessian $V_{\boldsymbol{\theta}^*}$,

(A3) $\mathcal{L}(Z, \boldsymbol{\theta})$ is differentiable at $\boldsymbol{\theta}^*$ almost surely (a.s.) or in probability,

(A4) $\mathcal{L}(Z, \boldsymbol{\theta})$ is Lipschitz near $\boldsymbol{\theta}^*$: $|\mathcal{L}(Z, \boldsymbol{\theta}_1) - \mathcal{L}(Z, \boldsymbol{\theta}_2)| \leq M(Z)\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$ with Lipschitz coefficient $M(Z)$ bounded in squared expectation, $\mathbb{E}[M(Z)^2] < \infty$.

Their first theorem states that under Assumption set 1, the machine-wise averaged estimator enjoys the same first-order statistical properties as the centralized solution.

Theorem 1 (Rosenblatt and Nadler (2016)) Under Assumption set 1, as $n \to \infty$ with p fixed, and any norm

$$\frac{\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|}{\|\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}^*\|} = 1 + o_P(1). \tag{2.13}$$

Definition 2.5.1 Two estimators are said to be *first-order equivalent* if their leading error terms converge to the same limit at the same rate, with the same limiting distribution.

Assumption set 1 implies that $\hat{\boldsymbol{\theta}}_n$ converges to $\boldsymbol{\theta}^*$ at rate $O(n^{-1/2})$ (Van der Vaart (2000), Corollary 5.53). Theorem 1 thus directly implies following corollary.

Corollary 1 (Rosenblatt and Nadler (2016)) The averaged estimator $\bar{\boldsymbol{\theta}}$ is first-order equivalent to the centralized solution $\hat{\boldsymbol{\theta}}_N$.

Remark 3.5.1 In practice, equation (2.12) is minimized approximately, typically by iterative approaches such as gradient descent (GD), stochastic gradient descent (SGD), etc. It is of great importance that Theorem 1 holds not only for the *exact* empirical minimizer $\hat{\boldsymbol{\theta}}_n$ but also for any *approximate* minimizer $\tilde{\boldsymbol{\theta}}_n$ satisfying $\hat{R}_n(\tilde{\boldsymbol{\theta}}_n) \leq \hat{R}_n(\hat{\boldsymbol{\theta}}_n) + o_P(n^{-1})$ (Van der Vaart (2000), Theorem 5.23). In other words, for the averaged minimizer to be first-order equivalent to the centralized solution, $o_P(n^{-1})$ precision is enough in minimizing $\hat{R}_N(\boldsymbol{\theta})$.
Optimality and robustness of $\bar{\boldsymbol{\theta}}$ are implications of Theorem 1 on its statistical properties. These properties are discussed by Rosenblatt and Nadler (2016), and we state them below. However, first let us point out the scope they claim Theorem 1 covers.

Scope. The learning tasks covered by Theorem 1 include: linear or nonlinear regression with l_2, Huber, or log likelihood loss; linear or nonlinear quantile regression with continuous predictors; binary regression; binary hinge loss regression (i.e. SVM) with continuous predictors, and unsupervised learning of location and scale. More significantly, it also covers regularized risk minimization $\boldsymbol{\theta}^* := arg\min_{\boldsymbol{\theta}}\{R(\boldsymbol{\theta}) + J(\boldsymbol{\theta})\}$ with a fixed regularization term $J(\boldsymbol{\theta})$, provided that the new loss function $\tilde{\mathcal{L}}(Z, \boldsymbol{\theta}) = \mathcal{L}(Z, \boldsymbol{\theta}) + J(\boldsymbol{\theta})$ satisfies the required assumptions. For further detail see Rosenblatt and Nadler (2016).

Some learning problems which are not covered by this theorem include: non-uniform allocation of samples to machines; non-convex parameter spaces; non-differentiable loss function with discrete predictors; the $n < p$ regime. In our setup from very first in this literature, we considered the linear regression model under the sparsity assumption. In this setup (under the assumptions we have made so far), Lee et al. (2017) showed that, it is still possible to devise a distributed scheme

that converges at the same rate as the centralized M-estimator. This is achievable by averaging *de-biased* machine-wise M-estimators not by simple average technique. On the other side, as shown by Shamir et al. (2014), in some cases the simple averaging may not be better than the estimator obtained from a single machine with n samples. However, since our problem setup satisfies all the assumptions required by theorem 1, it covers our problem, and we can use the theorem and its result in this study. Furthermore we will show that the estimation strategies we have used in our work, are both $\sqrt{n}$ and $\sqrt{N}$- consistent.

Optimality of averaging. Some notions of asymptotic optimality, such as Best Regular and Local Minimax depend only on leading order error term (Van der Vaart (2000), Chapter 8). Thus, if $\hat{\boldsymbol{\theta}}_N$ is optimal with respect to any of these criteria, equation (2.13) implies that so is $\bar{\boldsymbol{\theta}}$. Duchi et al. (2014) (Corollary 3) and in Liu and Ihler (2014), discussed an example when the loss function is negative log likelihood of the generative model. The centralized solution, being the maximum-likelihood estimate of $\boldsymbol{\theta}^*$, is optimal in several distinct senses. Given this, Theorem 1 implies that $\bar{\boldsymbol{\theta}}$ is optimal as well, and the the factor 1 cannot be improved.

Robustness. As it was discussed in section 2.2.3, there exist some cases where local machines suffer from potential outliers, therefore, it is required to be handled at the machine-level, aggregation level or both. Several methods to manage the issue have been proposed, a notable example was stated in section 2.2.3, and involved tackling the problem at the aggregation level. Hence, dealing with outliers at the machine level alone is sufficient if the possibility of a high proportion of outliers in any machine is small and machine failure is not a problem. Other scenarios call for the consideration of robust aggregation functions Minsker (2019).

Asymptotic Linearity. Asymptotic linearity of the estimator in some nonlinear transformation of the samples is essential in the proof of Theorem 1. This is known as the *Asymptotic Linearity* property, and the related transformation is the *Influence Function*. There are several other estimators with asymptotic linearity property, including L, R and Minimum Distance. Hence, first-order equivalence of averaging to the centralized solution is rather general. It typically holds for asymptotically Gaussian estimators (Rieder (2012), chapters 1 and 6) and has also been observed in other contexts, such as that of particle filters Achutegui et al. (2014).

Limiting distribution. The following limiting Gaussian distribution is provided right away by the asymptotic linearity of $\bar{\boldsymbol{\theta}}$ in the influence function.

Corollary 2 (Rosenblatt and Nadler (2016)) (Asymptotic Normality) Under the assumptions of Theorem 1, when $n \to \infty$ with p fixed, $\sqrt{N}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)$ converges in distribution to

$$\mathcal{N}(0, V_{\boldsymbol{\theta}^*}^{-1}\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\theta}^*)\nabla\mathcal{L}(\boldsymbol{\theta}^*)']V_{\boldsymbol{\theta}^*}^{-1}).$$

Corollary 2 enables us to construct confidence intervals and test hypotheses on the unknown $\boldsymbol{\theta}^*$. Furthermore, we need to estimate the asymptotic covariance matrix. To do this, any $O_P(N^{-1/2})$ consistent estimator of the covariance matrix will conserve the asymptotic normality according to Slutsky's theorem.

Second-order terms

Rosenblatt and Nadler (2016) empirically showed that, relatively little accuracy is lost when paral-

lelizing a linear model. However, much can be lost when the model is nonlinear. One reason is that the second-order error term may not be negligible. In our work, we have only focused on linear model, therefore, at first one would think that it is not necessary to discuss the second-order error term. But, Rosenblatt and Nadler (2016) showed that the second-order error term is imperative when deciding how many machines to deploy, as the first-order approximation of the error does not depend on K when N is fixed. The number of local machines to deploy is one of our major questions in the context of distributed learning when N is fixed. Subsequently, we bring the results obtained in Rosenblatt and Nadler (2016) to our work in order to address the question. Intuitively, the first-order term captures estimation variance which is reduced by averaging. The second-order term captures also bias, which is not reduced by averaging.
To study the second-order error term of $\bar{\boldsymbol{\theta}}$, they make suitable assumptions that ensure that *the machine-wise M-estimator* admits the following higher-order expansion:

$$\hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}^* + \xi_{-1/2}(\hat{\boldsymbol{\theta}}_n) + \xi_{-1}(\hat{\boldsymbol{\theta}}_n) + \xi_{-3/2}(\hat{\boldsymbol{\theta}}_n) + O_P(n^{-2}), \tag{2.14}$$

where $\xi_{-\alpha}(\hat{\boldsymbol{\theta}}_n)$ denotes the $O_P(n^{-\alpha})$ error term in $\hat{\boldsymbol{\theta}}_n$ and $\alpha = \{1/2, 1, 3/2, \dots\}$. It should be noted that $\xi_{-\alpha}(.)$ are data dependent themselves. Also note that this expansion is not specific for M-estimators, and they used this expansion for $\bar{\boldsymbol{\theta}}$ as well. For equation (2.14) to hold, the following set of assumptions with $s = 4$ is sufficient, see Rilstone et al. (1996).

Assumption set 2. There exist a neighborhood of $\boldsymbol{\theta}^*$ in which all of the following conditions hold:

(B1) Local differentiability: $\nabla^s \mathcal{L}(\boldsymbol{\theta}, Z)$ up to order s, exist a.s. and $\mathbb{E}[\|\nabla^s \mathcal{L}(\boldsymbol{\theta}^*, Z)\|] < \infty$.

(B2) Bounded empirical information: $(\nabla^2 \hat{R}_n(\boldsymbol{\theta}))^{-1} = O_P(1)$.

(B3) Lipschitz derivatives of order s:

$$\|\nabla^s \mathcal{L}(\boldsymbol{\theta}, Z) - \nabla^s \mathcal{L}(\boldsymbol{\theta}^*, Z)\| \leq M\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|,$$

where $\mathbb{E}[|M|] \leq C < \infty$.

Let us consider the notation used by Rosenblatt and Nadler (2016) and following Rilstone et al. (1996). Define a $(p \times 1)$ column vector $\boldsymbol{\delta}$, and $(p \times p)$ matrices $\gamma_0, \dots, \gamma_4$ as follows

$$\mathbb{E}[\xi_{-1}(\hat{\boldsymbol{\theta}}_n)] = n^{-1}\boldsymbol{\delta}; \quad \mathbb{E}[\xi_{-1}(\hat{\boldsymbol{\theta}}_n)]\mathbb{E}[\xi'_{-1}(\hat{\boldsymbol{\theta}}_n)] = n^{-2}\gamma_0 = n^{-2}\boldsymbol{\delta}\boldsymbol{\delta}';$$

$$\mathbb{E}[\xi_{-1/2}(\hat{\boldsymbol{\theta}}_n)\xi'_{-1/2}(\hat{\boldsymbol{\theta}}_n)] = n^{-1}\gamma_1; \quad \mathbb{E}[\xi_{-1}(\hat{\boldsymbol{\theta}}_n)\xi'_{-1/2}(\hat{\boldsymbol{\theta}}_n)] = n^{-2}\gamma_2; \tag{2.15}$$

$$\mathbb{E}[\xi_{-1}(\hat{\boldsymbol{\theta}}_n)\xi'_{-1}(\hat{\boldsymbol{\theta}}_n)] = n^{-2}\gamma_3 + o(n^{-2}); \quad \mathbb{E}[\xi_{-3/2}(\hat{\boldsymbol{\theta}}_n)\xi'_{-1/2}(\hat{\boldsymbol{\theta}}_n)] = n^{-2}\gamma_4 + o(n^{-2}).$$

2.4.3 Number of machines to deploy

As it was mentioned before, a key problem to be addressed in distributed setting is: How many local machines should the practitioner split the data over? to tackle the problem, two regimes can be considered: N fixed or n fixed. In Rosenblatt and Nadler (2016), section 6, both regimes are investigated. Let us point out their results and explore some detail.

Fixed n captures the local machine storage constraint: The total number of observations N is too large and there is always constraint on the memory limit for each machine. Hence, the practitioner needs to deploy more machines which allows processing more data at an obvious financial cost.

Fixed N captures either sampling or computational constraints: Sampling of a massive size N, obviously has cost on the operation in several senses. After collecting the sample data and tackling all the problems in the sampling step, the practitioner needs to distribute the observations across local computers in which there are computational constraints. The total sample size N is fixed, and it might take too long to process it on a single system. Splitting the data hence reduces run-time but also decreases the accuracy. In other words, by distributed setup, we trade accuracy for speed.

As per our problem setup, we consider fixed N. Interestingly, when N is fixed, by using the approximations and varying the number of machines K, we are able to see the accuracy-complexity tradeoff. A bound on K can be the solution to an optimization problem on the number of machines, with choosing either a desirable run-time or a desired error level as constraints for the problem. Rosenblatt and Nadler (2016) formulated the target functions for choosing the number of machines in two regimes mentioned above, however, we consider only the fixed N regime. To address the problem on the number of machines, when N is fixed, we wish to minimize runtime. Inspired by Shalev-Shwartz and Srebro (2008), we ask what is the *maximal* number of machines to minimize the runtime while a desired level of accuracy is maintained. According to Rosenblatt and Nadler (2016), the problem to be solved in order to choose the maximal value for K is given by

$$max\Big\{K \;\; s.t. \;\; \mathcal{E}(K) \leq \epsilon, N/K \; samples \; per \; machine\Big\}, \tag{2.16}$$

where $\mathcal{E}(K) := \mathbb{E}[\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_*\|^2]$, is the accuracy measure. Since in general there is no explicit or closed form for this quantity, we need to approximate it. As we are in the fixed-p regime, approximating the MSE by the asymptotic leading error term yields that this quantity is independent of K(Rosenblatt and Nadler (2016)). As it was shown in Rosenblatt and Nadler (2016), meaningful and interesting solutions to this optimization problem arise when we approximate $\mathcal{E}(K)$ by the second-order expression in the fixed-p regime. They provided the final form of the problem which is as follows

$$max\Big\{K \; s.t. \; ((K-1)K/N^2)Tr(\gamma_0) + (1/N)Tr(\gamma_1) + (K/N^2)Tr(\gamma_2 + \gamma_2' + \gamma_3 + \gamma_4 + \gamma_4') \leq \epsilon\Big\}.$$

To derive this formula for more complicated models, it takes formidable mathematical exercise. As a first application, let us consider using these formulas to the OLS estimator as an example.

Example(OLS). Given the standard generative linear model $\mathbf{Y} = \mathbb{X}'\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}$, with the explanatory variables satisfying $\mathbb{E}[\mathbb{X}] = 0$ and $Var[\mathbb{X}] = \Sigma$, and the noise $\boldsymbol{\varepsilon}$ independent of $\mathbb{X}$ with mean zero and $Var[\boldsymbol{\varepsilon}] = \sigma^2$. The loss is $\mathcal{L}(\mathbf{Y}, \mathbb{X}; \boldsymbol{\beta}) = \frac{1}{2}(\mathbf{Y} - \mathbb{X}'\boldsymbol{\beta})^2$, whose risk minimizer is the generative parameter, $\boldsymbol{\beta}^* = \boldsymbol{\beta}_0$. The following proposition, proved in Rosenblatt and Nadler (2016), provides explicit expressions for the second-order MSE matrix.

Proposition (Rosenblatt and Nadler (2016)) For the OLS problem, under the above generative linear model,

$$\gamma_0 = 0, \quad \gamma_1 = \sigma^2\Sigma^{-1}, \quad \gamma_2 = -(1+p)\sigma^2\Sigma^{-1}, \quad \gamma_3 = (1+p)\sigma^2\Sigma^{-1}, \quad \gamma_4 = (1+p)\sigma^2\Sigma^{-1}$$

Considering the example where they solved the optimization problem to obtain the maximal number of machines to deploy, let $N = 10^6$,$p = 10^3$, $\sigma^2 = 10$ and for simplicity $\Sigma = \mathbf{I}$ we have

$$\gamma_0 = 0, \quad \gamma_1 = 10\mathbf{I}, \quad \gamma_2 = -(1 + 10^3)10\mathbf{I}, \quad \gamma_3 = (1 + 10^3)10\mathbf{I}, \quad \gamma_4 = (1 + 10^3)10\mathbf{I},$$

substituting these values in the objective function $max\{K; \mathcal{E}(K) \leq \boldsymbol{\epsilon}\}$ mentioned before, the maximal number of machines to keep the MSE under 0.1 is $K \leq 8991$.

2.5 Averaging technique to be used for distributed Liu-type shrinkage estimation

Optimality of averaging is investigated by the studies to which we referred in previous sections. The problem setting for which the averaging is efficient covers the setting which is considered for the new class of shrinkage estimators introduced by Yüzbaşı et al. (2022). More importantly, as it was mentioned in section (1.6), Yüzbaşı et al. (2022) showed not only good asymptotic properties of these estimators, but also their efficiency using the MSE criteria. On the other side, in section (2.4), we noticed how averaging is efficient and optimal considering the MSE of estimators and investigating the optimality of averaging based on MSE of averaged answer. For these shrinkage estimators, the discussion, leads to make a straightforward decision that averaging can be used as an aggregation strategy in the central machine when the local machines use Liu-type shrinkage estimations. That is

$$\widehat{\boldsymbol{\beta}} = K^{-1} \sum_{j=1}^{K} \widehat{\boldsymbol{\beta}}_j$$

where $\widehat{\boldsymbol{\beta}}_j$ is the local estimate computed by machine j $(j = 1, \ldots, K)$ using listed estimators in section (1.6). There may be better strategies to merge local estimators in the central machine when a practitioner chooses the new class of shrinkage estimators for local machines. However, to be conservative, in our numerical study, we average the local estimates to compute the final estimate in the central machine. Furthermore, for this setup, Lee et al. (2017) showed that, it is still possible to devise a distributed scheme that converges at the same rate as the centralized M-estimator, and this is achievable by averaging *de-biased* machine-wise M-estimators not by simple average technique.

2.5.1 Asymptotic analysis of the Liu-type averaged estimator

In section (2.3), we defined the per-machine version of Liu-type shrinkage estimations introduced in section (1.6) for distributed sparse linear regression. Consistency of these estimators where they are used for global estimation, were investigated in Yüzbaşı et al. (2022). As it was determined before, in this study we use averaging technique to aggregate local estimators. Optimality of averaging was discussed in Liu and Ihler (2014), they also studied consistency of the averaged estimator. We mentioned their results in sections (1.6) and (2.4). In this section, we focus on asymptotic analysis of the averaged estimator when Liu-type shrinkage estimations are performed by local machines.

For $\sqrt{n}$ and $\sqrt{N}$-consistency of the averaged estimator, we desire the whole sample estimator to be $\sqrt{N}$-consistent. However, the whole sample is split into many (K) subsets, and the final answer is the average of local estimators. Hence, in order to investigate asymptotic properties

of the averaged estimator (the whole sample estimator), it is required to begin from asymptotic analysis of the local estimators. The asymptotic bias of an estimator for a sample of size N is defined as

$$\mathcal{B}(\widehat{\boldsymbol{\beta}}^*) = \mathbb{E}\big[\lim_{N\to\infty}\{\sqrt{N}(\widehat{\boldsymbol{\beta}}^* - \boldsymbol{\beta})\}\big],$$

and its asymptotic covariance is given by

$$\boldsymbol{\Gamma}(\widehat{\boldsymbol{\beta}}^*) = \mathbb{E}\big[\lim_{N\to\infty}\{N(\widehat{\boldsymbol{\beta}}^* - \boldsymbol{\beta})(\widehat{\boldsymbol{\beta}}^* - \boldsymbol{\beta})^T\}\big].$$

Consider Liu full model biased estimator (LFM)

$$\widehat{\boldsymbol{\beta}}^{LFM} = (\mathbb{X}^T\mathbb{X} + \mathbf{I}_p)^{-1}(\mathbb{X}^T\mathbb{X} + d\mathbf{I}_p)\widehat{\boldsymbol{\beta}}^{LSE},$$

where $0 < d < 1$ is the biasing parameter. Furthermore, consider the following regularity conditions mentioned in Yüzbaşı et al. (2022) to evaluate the asymptotic properties of the estimators.

Regularity conditions

1. $\frac{1}{N}\max_{1\le i\le N}\mathbf{x}_i^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbf{x}_i \to 0$ as $N\to\infty$, where $\mathbf{x_i^T}$ is the ith row of $\mathbb{X}$.
2. $\lim_{N\to\infty} N^{-1}(\mathbb{X}^T\mathbb{X}) = \lim_{N\to\infty}\mathbb{C}_N = \mathbb{C}$, where $\mathbb{C}$ is a finite and positive definite matrix.
3. $\lim_{N\to\infty}\mathbb{F}_N(d) = \mathbb{F}_d$, for finite $\mathbb{F}_d$ where $\mathbb{F}_N(d) = (\mathbb{C}_N+\mathbf{I}_p)^{-1}(\mathbb{C}_N+d\mathbf{I}_p)$ and $\mathbb{F}_d = (\mathbb{C}+\mathbf{I}_p)^{-1}(\mathbb{C}+d\mathbf{I}_p)$.

If the whole dataset were to be analysed by a single computer, the global estimator would possess the property stated in the following theorem under the above regularity conditions.

Theorem 3.9(Yüzbaşı et al. (2022)) If the data is not distributed, $\mathbb{C}$ is non-singular, and $0 < d < 1$, then as $N\to\infty$ we have

$$\sqrt{N}\Big(\widehat{\boldsymbol{\beta}}^{LFM} - \boldsymbol{\beta}\Big) \xrightarrow{\text{d}} \mathcal{N}_p\Big(-(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta},\ \sigma^2\mathbb{S}\Big),$$

where $\mathbb{S} = \mathbb{F}_d\mathbb{C}^{-1}\mathbb{F}_d^T$, $\widehat{\boldsymbol{\beta}}^{LFM}$ is the global estimator, $\mathcal{N}(.,.)$ denotes multivariate normal distribution, subscript p is the length of $\boldsymbol{\beta}$, and $(\xrightarrow{\text{d}})$ means convergence in distribution.

Now for the local estimators, similar to the whole sample estimator (global estimator), the asymptotic bias and covariance of the local estimators can be defined as follows,

$$\mathcal{B}(\widehat{\boldsymbol{\beta}}_j^*) = \mathbb{E}\big[\lim_{n\to\infty}\{\sqrt{n}(\widehat{\boldsymbol{\beta}}_j^* - \boldsymbol{\beta})\}\big] \quad \text{and} \quad \boldsymbol{\Gamma}(\widehat{\boldsymbol{\beta}}_j^*) = \mathbb{E}\big[\lim_{n\to\infty}\{n(\widehat{\boldsymbol{\beta}}_j^* - \boldsymbol{\beta})(\widehat{\boldsymbol{\beta}}_j^* - \boldsymbol{\beta})^T\}\big].$$

In our problem, rows of the design matrix $\mathbb{X}_{N\times p}$ are split into K distinct submatrices randomly and evenly. Therefore, dimensions of each submatrix $\mathbb{X}_j$ is $(n\times p)$ and it is obvious that $n = \frac{N}{K}$. In order to perform asymptotic analysis of the local estimators, it is required to restate regularity conditions. Prior to that, as $n = \frac{N}{K}$, when $N\to\infty$, for K fixed it implies that $n\to\infty$ as well. Now, for an arbitrary subsample S_j, given that $N\to\infty$ implies $n\to\infty$, one may immediately consider the following regularity conditions as a result of conditions mentioned before,

Local regularity conditions

1. $\frac{1}{n}\max_{1\le i\le n}\mathbf{x}_i^T(\mathbb{X}_j^T\mathbb{X}_j)^{-1}\mathbf{x}_i \to 0$ as $n\to\infty$, where $\mathbb{X}_j$ is the design matrix in machine j, $j\in\{1,\ldots,K\}$, and $\mathbf{x_i^T}$ is the ith row of $\mathbb{X}_j$.
2. $\lim_{n\to\infty} n^{-1}(\mathbb{X}_j^T\mathbb{X}_j) = \lim_{n\to\infty}\mathbb{C}_n = \mathbb{C}$, where $\mathbb{C}$ is a finite and positive definite matrix.
3. $\lim_{n\to\infty}\mathbb{F}_n(d) = \mathbb{F}_d$, for finite $\mathbb{F}_d$ where $\mathbb{F}_n(d) = (\mathbb{C}_n+\mathbf{I}_p)^{-1}(\mathbb{C}_n+d\mathbf{I}_p)$ and $\mathbb{F}_d = (\mathbb{C}+\mathbf{I}_p)^{-1}(\mathbb{C}+d\mathbf{I}_p)$.

Corollary 2.5.1 Under the local regularity conditions, theorem (3.9) implies that for each local estimator, $\widehat{\boldsymbol{\beta}}_j^{LFM}$, as $n\to\infty$ we have,

$$\forall j\in\{1,\ldots,K\},\quad \sqrt{n}\Big(\widehat{\boldsymbol{\beta}}_j^{LFM}-\boldsymbol{\beta}\Big)\xrightarrow{\mathrm{d}}\mathcal{N}_p\Big(-(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta},\ \sigma^2\mathbb{S}\Big).$$

Moreover, consider forming asymptotic bias of the averaged estimator $\widehat{\boldsymbol{\beta}}^{LFM}$,

$$\frac{1}{K}\left(\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LFM}-\boldsymbol{\beta})\Big]+\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}_2^{LFM}-\boldsymbol{\beta})\Big]+\cdots+\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}_K^{LFM}-\boldsymbol{\beta})\Big]\right)$$

$$=\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}\frac{1}{K}\Big((\widehat{\boldsymbol{\beta}}_1^{LFM}-\boldsymbol{\beta})+(\widehat{\boldsymbol{\beta}}_2^{LFM}-\boldsymbol{\beta})+\cdots+(\widehat{\boldsymbol{\beta}}_K^{LFM}-\boldsymbol{\beta})\Big)\Big]$$

$$=\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}\Big((\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM})-\boldsymbol{\beta}\Big)\Big]$$

$$=\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}^{LFM}-\boldsymbol{\beta})\Big]$$

where $\widehat{\boldsymbol{\beta}}^{LFM}=\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM}$ is the averaged estimator. This leads us to investigate $\sqrt{n}$-consistency of the averaged estimator.

Theorem 2.5.1 If $0<d<1$ and $\mathbb{C}$ is non-singular, then as $n\to\infty$

$$\sqrt{n}\Big(\widehat{\boldsymbol{\beta}}^{LFM}-\boldsymbol{\beta}\Big)\xrightarrow{\mathrm{d}}\mathcal{N}\Big(-(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta},\ \frac{1}{K}\sigma^2\mathbb{S}\Big),$$

where $\widehat{\boldsymbol{\beta}}^{LFM}=\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM}$.

Proof. Since $\widehat{\boldsymbol{\beta}}^{LFM}$ is a linear function of $\widehat{\boldsymbol{\beta}}_j^{LFM}$ $(j=1,\ldots,K)$, according to Conclusion (1), $\widehat{\boldsymbol{\beta}}^{LFM}$ is asymptotically normally distributed.

The asymptotic bias of $\widehat{\boldsymbol{\beta}}^{LFM}$ is obtained as

$$\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}^{LFM}-\boldsymbol{\beta})\Big]=\mathbb{E}\Big[\lim_{n\to\infty}\sqrt{n}\Big((\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM})-\boldsymbol{\beta}\Big)\Big]$$

$$=\frac{1}{K}\mathbb{E}\Big[\sum_{j=1}^{K}\lim_{n\to\infty}\sqrt{n}(\widehat{\boldsymbol{\beta}}_j^{LFM}-\boldsymbol{\beta})\Big]$$

$$=\frac{1}{K}\sum_{j=1}^{K}-(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta}$$

$$=-(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta},$$

since local estimators are independent, the asymptotic covariance is

$$\boldsymbol{\Gamma}(\widehat{\boldsymbol{\beta}}^{LFM})=\boldsymbol{\Gamma}\Big(\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM}\Big)$$

$$=\frac{1}{K^2}\sum_{j=1}^{K}\boldsymbol{\Gamma}(\widehat{\boldsymbol{\beta}}_j^{LFM})$$

$$=\frac{1}{K^2}\sum_{j=1}^{K}\sigma^2\mathbb{S}$$

$$=\frac{1}{K}\sigma^2\mathbb{S}.$$

□

Moreover, let us see if $\sqrt{N}\Big(\widehat{\boldsymbol{\beta}}^{LFM}-\boldsymbol{\beta}\Big)$ is bounded in probability as well. Subsequently, see how closer the estimate can be to the true value of $\boldsymbol{\beta}$. Therefore, $\sqrt{N}$-consistency of the averaged estimator can be investigated.

Proposition 2.5.2 If K is fixed, $0<d<1$ and $\mathbb{C}$ is non-singular, then as $N\to\infty$

$$\sqrt{N}\Big(\widehat{\boldsymbol{\beta}}^{LFM}-\boldsymbol{\beta}\Big)\xrightarrow{\mathrm{d}}\mathcal{N}\Big(-\sqrt{K}(1-d)(\mathbb{C}+\mathbf{I}_p)^{-1}\boldsymbol{\beta},\ \frac{1}{K}\sigma^2\mathbb{S}\Big),$$

where $\widehat{\boldsymbol{\beta}}^{LFM}=\dfrac{1}{K}\sum\limits_{j=1}^{K}\widehat{\boldsymbol{\beta}}_j^{LFM}$.

Proof. First, it should be noted that,

$$N \to \infty \implies (nK) \to \infty \xrightarrow{K \text{ is a fixed number}} n \to \infty \;\; s.t. \;\; \frac{n}{N} \to \frac{1}{K}, \tag{2.17}$$

thus, $n \to \infty$ allows us to use Conclusion (1) in our proof as it was for $\sqrt{n}$-consistency.
As the averaged estimator $\widehat{\boldsymbol{\beta}}^{LFM}$, is a linear function of $\widehat{\boldsymbol{\beta}}_j^{LFM}$ $(j = 1, \ldots, K)$, it is asymptotically normally distributed by making use of (2.17) and Conclusion (1). The asymptotic bias is obtained as follows

$$\mathbb{E}\Big[\lim_{N\to\infty} \sqrt{N}(\widehat{\boldsymbol{\beta}}^{LFM} - \boldsymbol{\beta})\Big] \overset{(2.17)}{=} \mathbb{E}\Big[\lim_{n\to\infty} \sqrt{nK}\Big((\frac{1}{K}\sum_{j=1}^{K} \widehat{\boldsymbol{\beta}}_j^{LFM}) - \boldsymbol{\beta}\Big)\Big]$$

$$= \frac{\sqrt{K}}{K}\mathbb{E}\Big[\sum_{j=1}^{K} \lim_{n\to\infty} \sqrt{n}(\widehat{\boldsymbol{\beta}}_j^{LFM} - \boldsymbol{\beta})\Big]$$

$$= \frac{1}{\sqrt{K}}\sum_{j=1}^{K} -(1-d)(\mathbb{C} + \mathbf{I}_p)^{-1}\boldsymbol{\beta}$$

$$= -\sqrt{K}(1-d)(\mathbb{C} + \mathbf{I}_p)^{-1}\boldsymbol{\beta},$$

and asymptotic covariance

$$\boldsymbol{\Gamma}(\widehat{\boldsymbol{\beta}}^{LFM}) = \boldsymbol{\Gamma}\Big(\frac{1}{K}\sum_{j=1}^{K} \widehat{\boldsymbol{\beta}}_j^{LFM}\Big)$$

$$= \frac{1}{K^2}\sum_{j=1}^{K} \Gamma(\widehat{\boldsymbol{\beta}}_j^{LFM})$$

$$= \frac{1}{K^2}\sum_{j=1}^{K} \sigma^2\mathbb{S}$$

$$= \frac{1}{K}\sigma^2\mathbb{S}.$$

It is of great importance to mention that in the above expressions for the asymptotic covariance, we used the the fact that, when K is fixed, $N \to \infty$ implies $n \to \infty$ such that $\frac{n}{N} \to \frac{1}{K}$. This is the reason why we could make use of $\Gamma(\widehat{\boldsymbol{\beta}}_j^{LFM}) = \sigma^2\mathbb{S}$ which was true when $n \to \infty$ in Theorem 1. □

To verify our results on the asymptotic analysis of the $\widehat{\boldsymbol{\beta}}^{LFM}$ estimator, a simulation study is conducted in Chapter (3). Furthermore, as for the asymptotic analysis of $\widehat{\boldsymbol{\beta}}^{RFM}$, $\widehat{\boldsymbol{\beta}}_1^{RFM}$, $\widehat{\boldsymbol{\beta}}_1^{RSM}$, $\widehat{\boldsymbol{\beta}}_1^{LFM}$, $\widehat{\boldsymbol{\beta}}_1^{LSM}$, $\widehat{\boldsymbol{\beta}}_1^{LPT}$, $\widehat{\boldsymbol{\beta}}_1^{LS}$, and $\widehat{\boldsymbol{\beta}}_1^{LPS}$ we have shown the results through the simulation study.

Chapter 3

Simulation study

3.1 Introduction

To implement distributed statistical learning (inference), particularly, evaluate performance of the shrinkage estimators discussed throughout this literature, we conduct a Monte Carlo simulation study using the statistical software R. We illustrate performance of the new class of shrinkage estimators proposed in Yüzbaşı et al. (2022) and discussed in section (2.3.2) in the context of distributed sparse linear regression analysis.

Consider following sparse linear model to generate the response vector $\mathbf{Y}$,

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where the design matrix $\mathbb{X}_{N\times p}$ is generated from a multivariate normal distribution with mean zero and covariance $\Sigma_{p\times p}$, and $\boldsymbol{\varepsilon}_{N\times 1}$ follows i.i.d. $\mathcal{N}(0,1)$. Condition number (CN) of a matrix is defined as the ratio of its largest eigen value to its smallest eigenvalue. CN of $\mathbb{X}^T\mathbb{X}$ is considered to assess multicollinearity of independent variables. Therefore, in each machine the condition number of $\mathbb{X}_j^T\mathbb{X}_j$ is used for multicollinearity of the allocated data to the local machine M_j. It was suggested by Fan and Li (2001) that the data has multicollinearity if the CN value is larger than 30. The total sample size is fixed to be $N = 10^4$, and elements of the covariance matrix are $\sigma_{i,j} = \rho^{|i-j|}$ where $\rho \in \{0.3, 0.6\}$. Regression coefficients are set as $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T = (\boldsymbol{\beta}_1^T, \mathbf{0}_{p_2}^T)^T$ with $\boldsymbol{\beta}_1 = (\underbrace{1, \ldots, 1}_{p_1})^T$. Number of predictor variables are $(p_1, p_2) = (5, 15)$. Generated training dataset under above mentioned assumptions is distributed across K machines randomly and evenly. As per our suggestion, the estimators $\widehat{\boldsymbol{\beta}}^{LFM}$, $\widehat{\boldsymbol{\beta}}^{RFM}$, $\widehat{\boldsymbol{\beta}}_1^{RFM}$, $\widehat{\boldsymbol{\beta}}_1^{RSM}$, $\widehat{\boldsymbol{\beta}}_1^{LFM}$, $\widehat{\boldsymbol{\beta}}_1^{LSM}$, $\widehat{\boldsymbol{\beta}}_1^{LPT}$, $\widehat{\boldsymbol{\beta}}_1^{LS}$, and $\widehat{\boldsymbol{\beta}}_1^{LPS}$ are considered to be used by local machines. Performance of the estimators is illustrated as the number of machines, K, varies in $\{1, 10, 20, 50, 100, 200, 500\}$ for a fixed N. MSE is the evaluation criteria when these estimators are used for distributed analysis. Moreover, to see boundedness of $\sqrt{n}(\bar{\boldsymbol{\beta}} - \boldsymbol{\beta})$ when $n \to \infty$ and $\sqrt{N}(\bar{\boldsymbol{\beta}} - \boldsymbol{\beta})$ when $N \to \infty$, we conduct a numerical experiment to evaluate respective theoretical results. We have also considered a real data example to assess performance of the shrinkage estimations mentioned in this literature.

3.2 Performance of the aggregated estimator with respect to the number of machines

The experiment is repeated 100 times to see the result over several replications. The final result is the average taken over all those replications. In each replication, the whole dataset is generated with the setting mentioned above, and is split into K subsets (local samples) randomly and evenly. Let $\mathbb{S} = \{1, \ldots, N\}$ be the index set of whole sample and $S_j (j = 1, \ldots, K)$ denote the index set of local samples with $S_i \cap S_j = \emptyset$ for any $i \neq j$. Next, for each S_j, an estimate $\boldsymbol{\beta}_j^{\blacktriangle}$ is computed where $\boldsymbol{\beta}_j^{\blacktriangle}$ is one of the estimation strategies discussed before. Thus, there would be K local estimates $\boldsymbol{\beta}_j^{\blacktriangle}$'s to be aggregated as a final estimate. To do this, as it was determined before, we use averaging technique i.e.,

$$\widehat{\boldsymbol{\beta}} = \bar{\boldsymbol{\beta}} = \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{\beta}_j^{\blacktriangle}$$

Lastly, we need to observe performance of each estimation strategy with respect to the number of data partitions (machines), K. In each replication for different values of K, MSE of the estimates are saved in rows of a matrix. First row of the matrix is considered to include MSE of the estimator when $K = 1$ (global estimator). For $K = 10$ the dataset (observations) is divided into ten subsets, based on each subset a local estimate is computed and the final estimate is the average of these ten local estimates. MSE of the final estimate is saved in a row of the matrix, next row contains MSE of the final estimate when $K = 20$, and we continue this to the last row which is for $K = 500$. Columns of the matrix are considered to save results in each replication. As there are 100 replications, it is clear that the matrix has 7 rows containing MSE's for each K and 100 columns containing the results from each replication.

3.2.1 Observations and result

In order to see the performance of the aggregated estimators, we have plotted MSE of the aggregated estimators with respect to the number of machines. Figures (3.1) and (3.2) are the illustrations of MSE values based on the generated data with $\rho = 0.6$ and $\rho = 0.3$ respectively. To be more clear about the plots, let us provide more detail about the simulation steps toward creating the plots. For example when $K = 20$ its respective point is obtained as follows

$$\left.\begin{array}{r} M_1 \to \boldsymbol{\beta}_1^{\blacktriangle} \\ M_2 \to \boldsymbol{\beta}_2^{\blacktriangle} \\ \vdots \\ M_{20} \to \boldsymbol{\beta}_{20}^{\blacktriangle} \end{array}\right\} \longrightarrow \quad \frac{1}{20} \sum_{j=1}^{20} \boldsymbol{\beta}_j^{\blacktriangle} = \widehat{\boldsymbol{\beta}} \ : \texttt{final estimate} \longrightarrow$$

$$\left.\begin{array}{r} MSE_1(\widehat{\boldsymbol{\beta}}) \\ MSE_2(\widehat{\boldsymbol{\beta}}) \\ \vdots \\ MSE_{100}(\widehat{\boldsymbol{\beta}}) \end{array}\right\} \longrightarrow \quad \frac{1}{100} \sum_{s=1}^{100} MSE_s(\widehat{\boldsymbol{\beta}}) = MSE(\widehat{\boldsymbol{\beta}}) : \texttt{averaged mse over 100 replications}$$

and $MSE(\widehat{\boldsymbol{\beta}})$ is MSE of the respective estimator indicated by a point in the plots for $K = 20$ on the K axis. It is obvious that MSE of any estimation strategy should grow when the number of machines (K) increases. This is the trend which is observable when we look at the plots, and it shows that the way we implemented the distributed analysis has worked properly.

From figure (3.2) and (3.1) we can see that,

1. The results when $\rho = 0.3$ are more satisfactory than those when $\rho = 0.6$, which is due to the less correlation among the covariates when $\rho = 0.3$. However, the difference in their performance is not significant and this shows that the local estimators have been able to combat multicollinearty even though they have only access to subsets of the data.

2. MSE of the aggregated estimators as it is illustrated in (3.2) and (3.1) does not increase significantly even when the data is split up to many chunks.

We have also considered Ridge estimator in our simulation study to be compared with the other shrinkage estimators we have used in this literature. A notable work on performing distributed ridge regression was conducted by Dobriban and Sheng (2020). They showed that the ridge estimator is affected very little even though the data is split up into many parts. Following their algorithm, we have performed the ridge regression on our generated dataset to visualize a comparison with the new class of Liu-type shrinkage estimators. As it is shown in the plots, this class of shrinkage estimators have comparable performance with ridge on the same dataset and under the same conditions in terms of distributed regression. In their paper, they proposed an *optimal weighted average algorithm* to aggregate local estimates. However, we have used simple average method in our study because we have considered a standard architecture of distributed learning in which the data is distributed evenly and randomly. Its asymptotic properties was also investigated in previous chapters. Moreover, the simple average method is the simplest way in terms of computation complexity, communication costs and algorithm run-time.

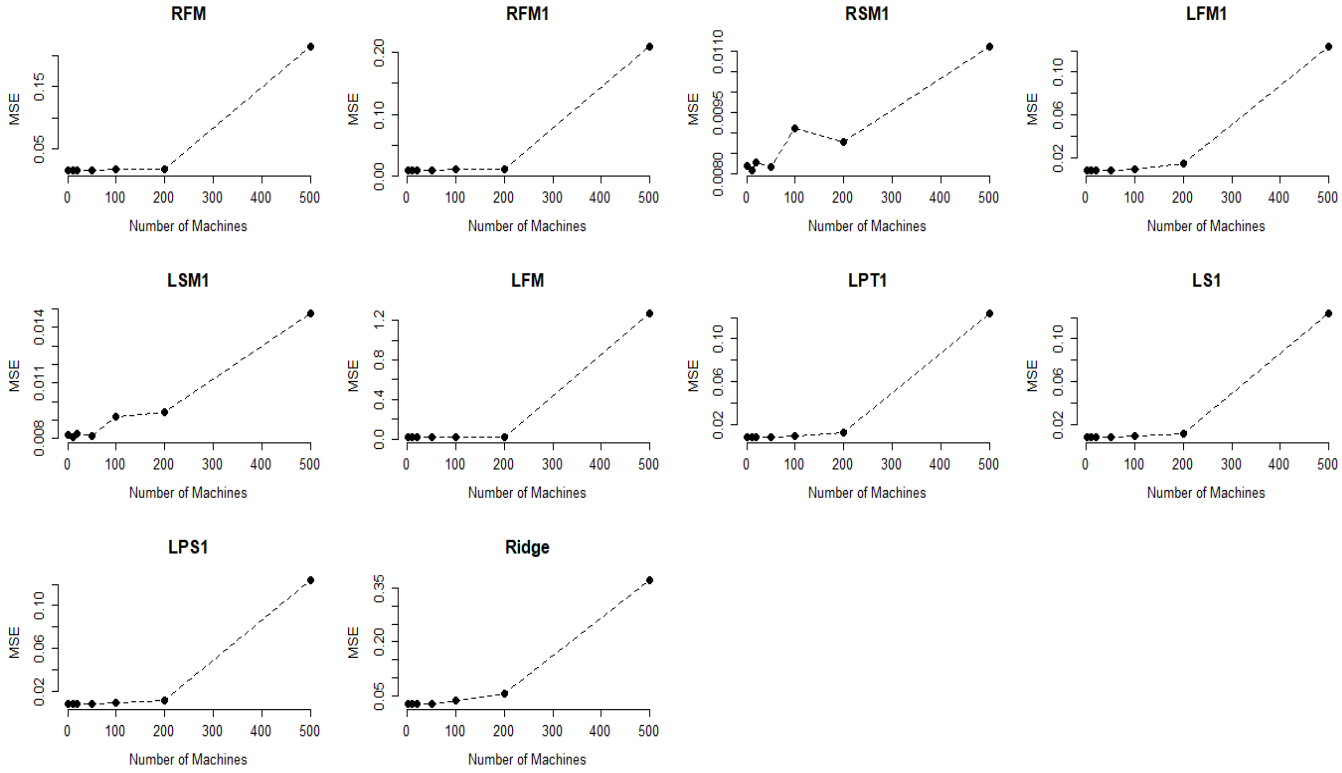

Fig. 3.1 *MSE of the aggregated estimators when* $\rho = 0.6$.

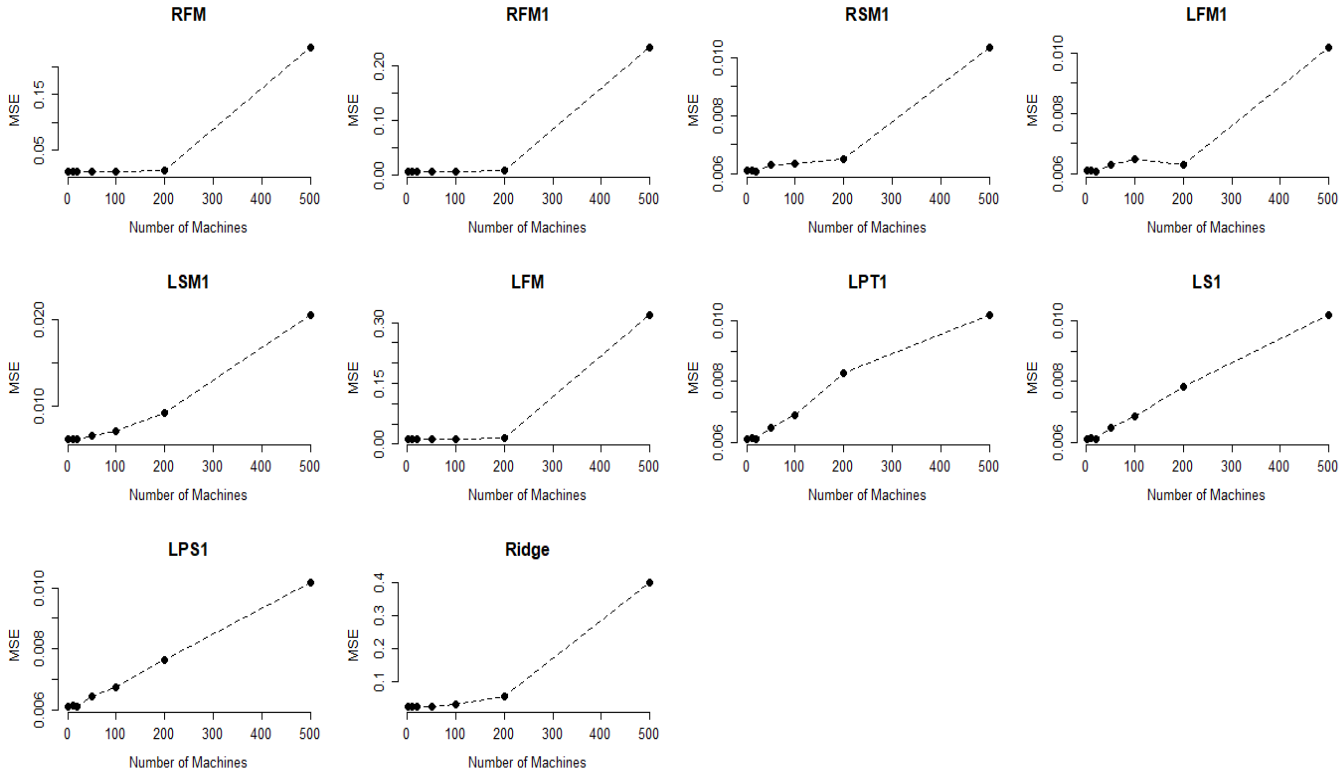

Fig. 3.2 *MSE of the aggregated estimators when* $\rho = 0.3$.

In our simulation, as we are considering a problem which is not a real one, significant and insignificant covariates are already set. For example, in our simulation we set them as $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T = (\boldsymbol{\beta}_1^T, \mathbf{0}_{p_2}^T)^T$ with $\boldsymbol{\beta}_1 = (\underbrace{1, \ldots, 1}_{p_1})^T$. Hence, we have prior knowledge that tells us to fit a sub-model, with first five covariates that are significantly important in explaining the response and other fifteen are insignificant. However, in application, if we do not have prior knowledge on data, one can perform stepwise or other variable selection techniques to select the best subset. This is what we have done when we study a real dataset in section (3.4).

3.3 Consistency and asymptotic normality of the aggregated estimator

In the following sections we have studied consistency and asymptotic normality of the aggregated estimators for the beforementioned Liu-type shrinkage strategies. The data is generated under the same setting we considered in the introduction section (3.1).

3.3.1 Consistency

In order to show that the aggregated estimator $\widehat{\boldsymbol{\beta}}$ is $\sqrt{n}$ and $\sqrt{N}$-consistent, it must be that, $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = O_p(1)$ and $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = O_p(1)$ respectively. That is, in our numerical study, we need to illustrate that, $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ and $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ are bounded in probability. They should thus be bounded when n and N grow.
Therefore, let us consider N to choose its values in $\{2000, 4000, 7000, 10000, 15000\}$, for fixed $K = 50$, and n subsequently, to vary in $\{40, 80, 140, 200, 300\}$. In the box plots we have illustrated behaviour of the abovementioned expressions over 50 replications. In figure (3.3), $\|\sqrt{n}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})\|_2$ values are plotted versus $n \in \{40, 80, 140, 200, 300\}$, where $\widehat{\boldsymbol{\beta}}^{\blacktriangle}$ is one of the before listed aggregated estimators, and n is the sample size in each machine. Thus, for a given n on the x axis, each box along with its whiskers and outer data points represent $\|\sqrt{n}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})\|_2$ for the respective aggregated estimator over 50 replications. As it evident from the plots, the values do not explode and $\|\sqrt{n}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})\|_2$ are bounded when n grows. This shows approximate $\sqrt{n}$-consistency of the aggregated estimators.

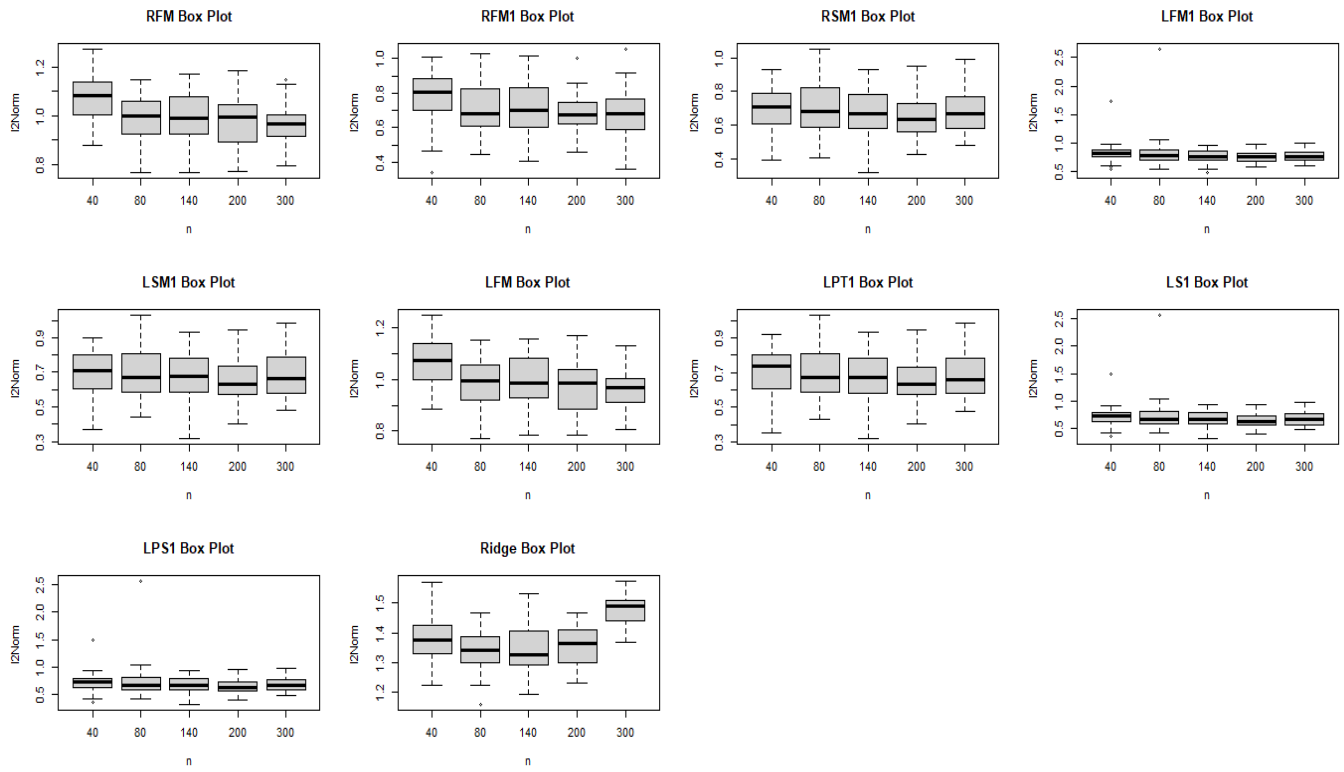

Fig. 3.3 *Behaviour of* $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ *when* n *grows.*

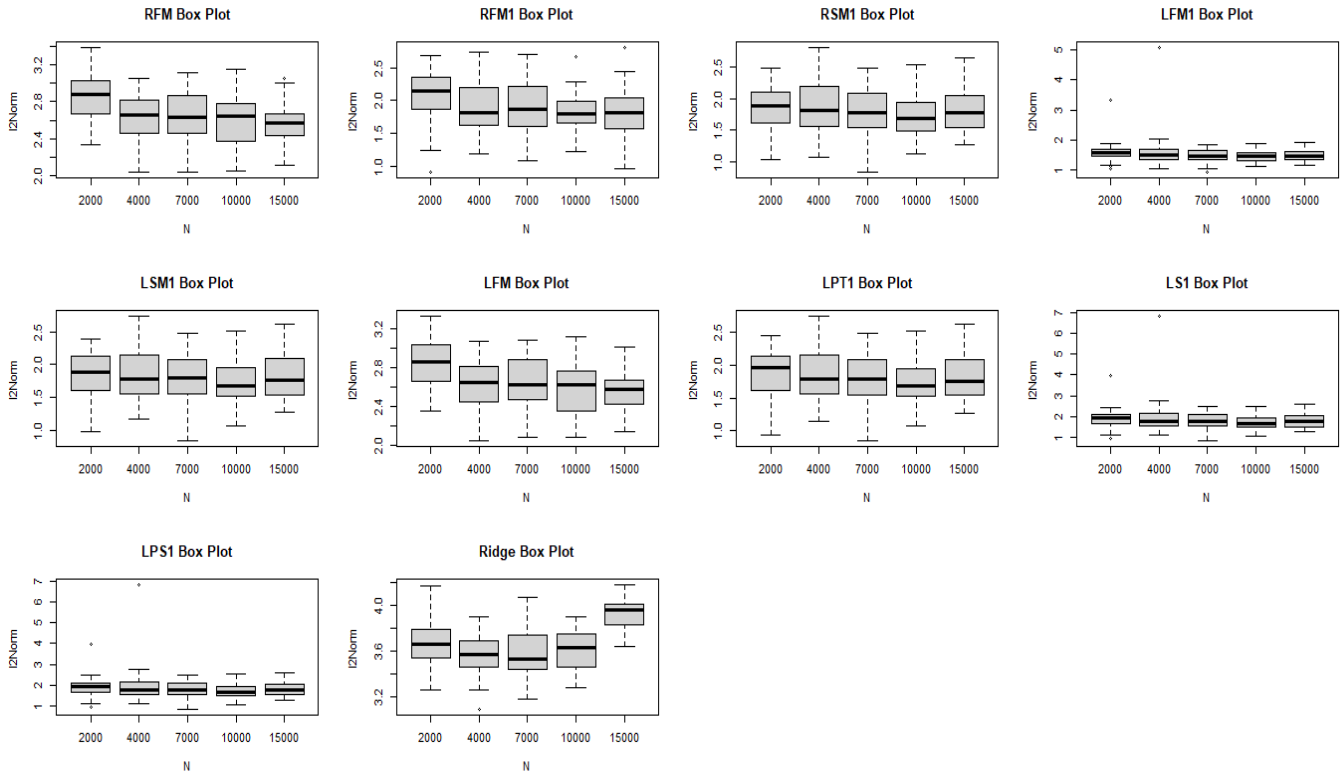

Fig. 3.4 *Behaviour of* $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ *when* N *grows.*

Similarly, for the N-consistency, in figure (3.4) we have plotted $\|\sqrt{N}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})\|_2$ values versus $N \in \{2000, 4000, 7000, 10000, 15000\}$, where N is the total sample size. For a given N on the x axis, each box along with its whiskers and outer data points represent $\|\sqrt{N}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})\|_2$ for the respective aggregated estimator over 50 replications. Again, the values are in a bounded range, and this illustrates approximate $\sqrt{N}$-consistency of the aggregated estimators which is a stronger result than that of $\sqrt{n}$-consistency.

3.3.2 Asymptotic normality

In order to show asymptotic normality, first let us mention a property of a random vector $\mathbf{X}$ having a multivariate normal distribution which is proved in Johnson et al. (2002).

Result 4.2.(Johnson et al. (2002)) If $\mathbf{X}$ is distributed as $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (multivariate normal distribution), then any linear combination of variables $\mathbf{a}'\mathbf{X} = a_1X_1 + a_2X_2 + \cdots + a_pX_p$ is distributed as $N(\mathbf{a}'\boldsymbol{\mu}, \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a})$ (univariate normal distribution). Also, if $\mathbf{a}'\mathbf{X}$ is distributed as $N(\mathbf{a}'\boldsymbol{\mu}, \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a})$ for every $\mathbf{a}$, then $\mathbf{X}$ must be $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Remark 3.3.2.1. An immediate implication of the abovementioned result is that, $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if every linear combination of the components of $\mathbf{X}$ follows univariate normal distribution.

Hence, in order to investigate asymptotic normality of $\mathbf{V} = (v_1, v_2, \ldots, v_p)^T = \sqrt{n}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})$, with $p = p_1 + p_2 = 20$, we need to study univariate normality of any linear combination of this vector. For example, we have considered three linear combinations that extract v_1, v_3 and v_5 elements of the vector $\mathbf{V}$. This setting is considered for $\mathbf{V} = (v_1, v_2, \ldots, v_p)^T = \sqrt{N}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})$ as well. Followings are Q-Q plots to illustrate normality of v_1, v_3, and v_5 for $n = 40$, $n = 140$, and $n = 300$ over 50 replications.

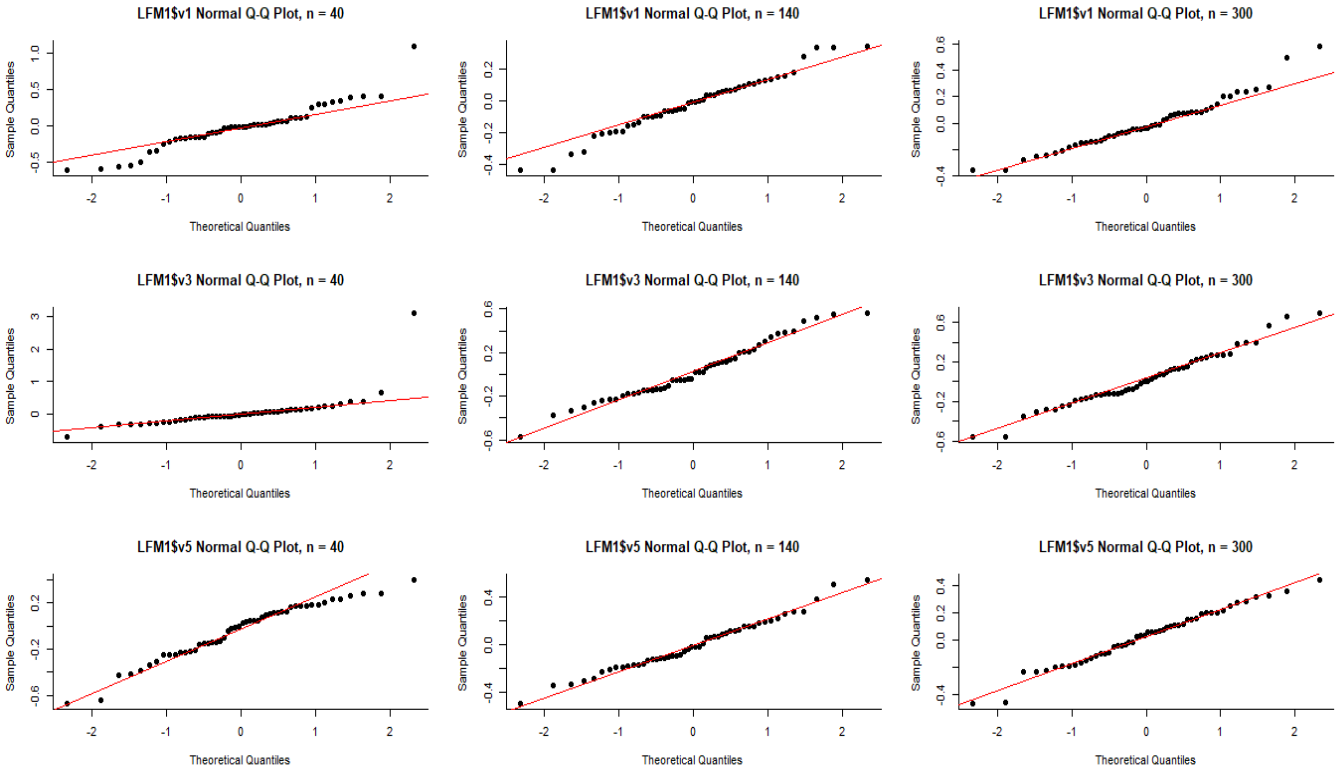

Fig. 3.5 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LFM} - \boldsymbol{\beta})$ *and* n *grows.*

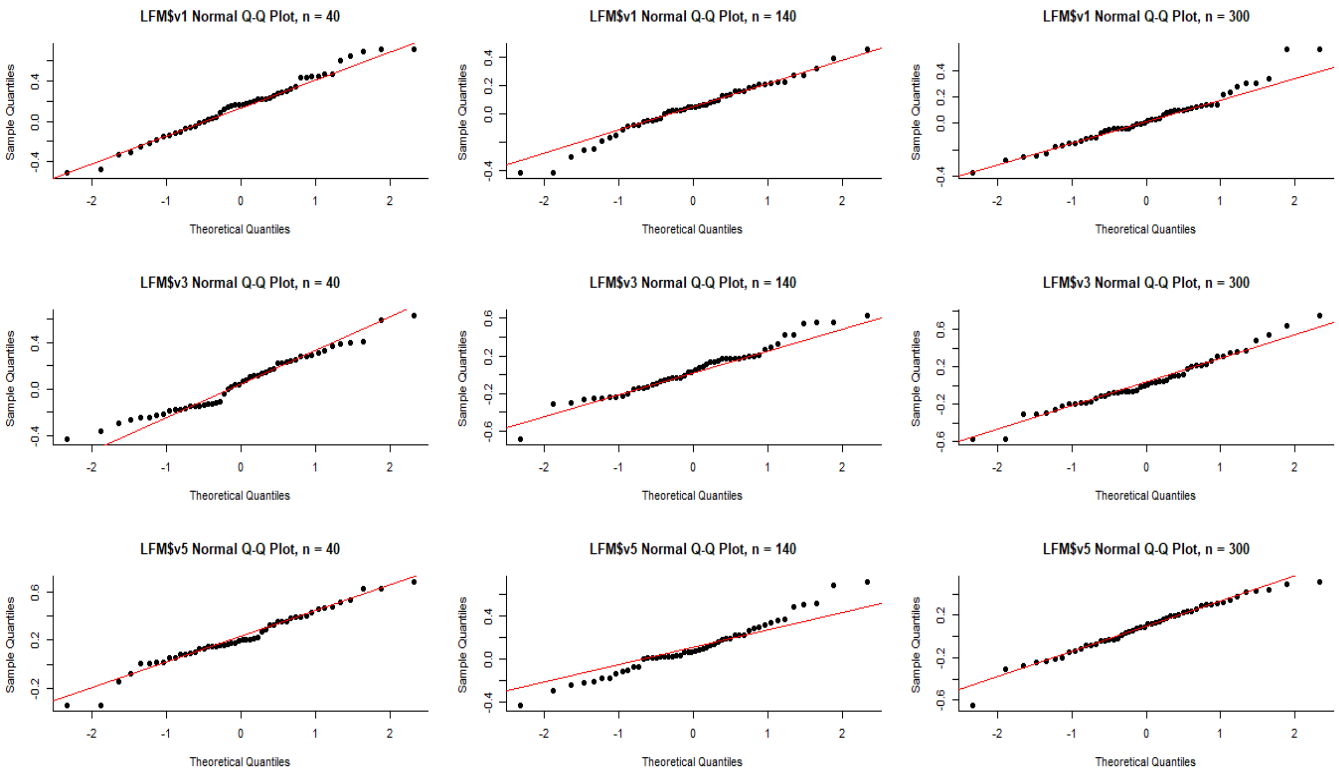

Fig. 3.6 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}^{LFM} - \boldsymbol{\beta})$ *and* n *grows.*

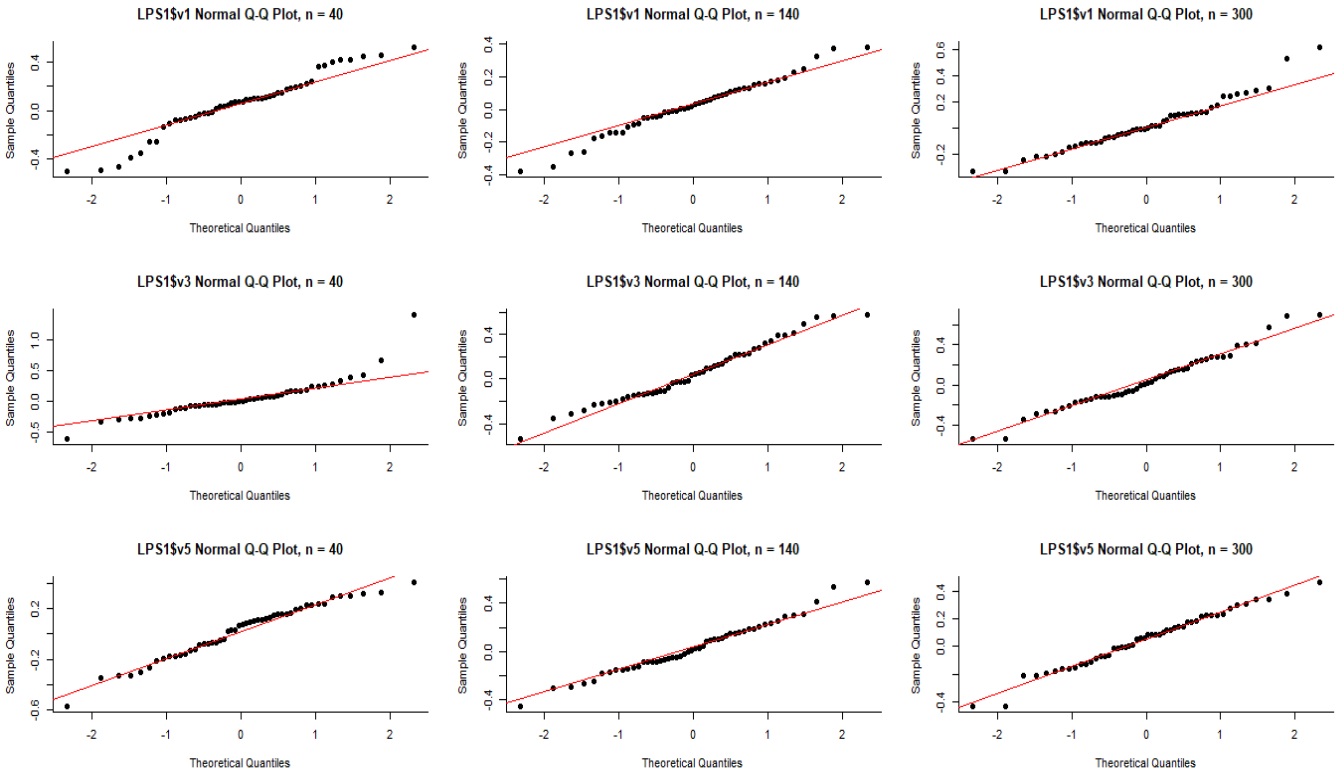

Fig. 3.7 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LPS} - \boldsymbol{\beta})$ *and* n *grows.*

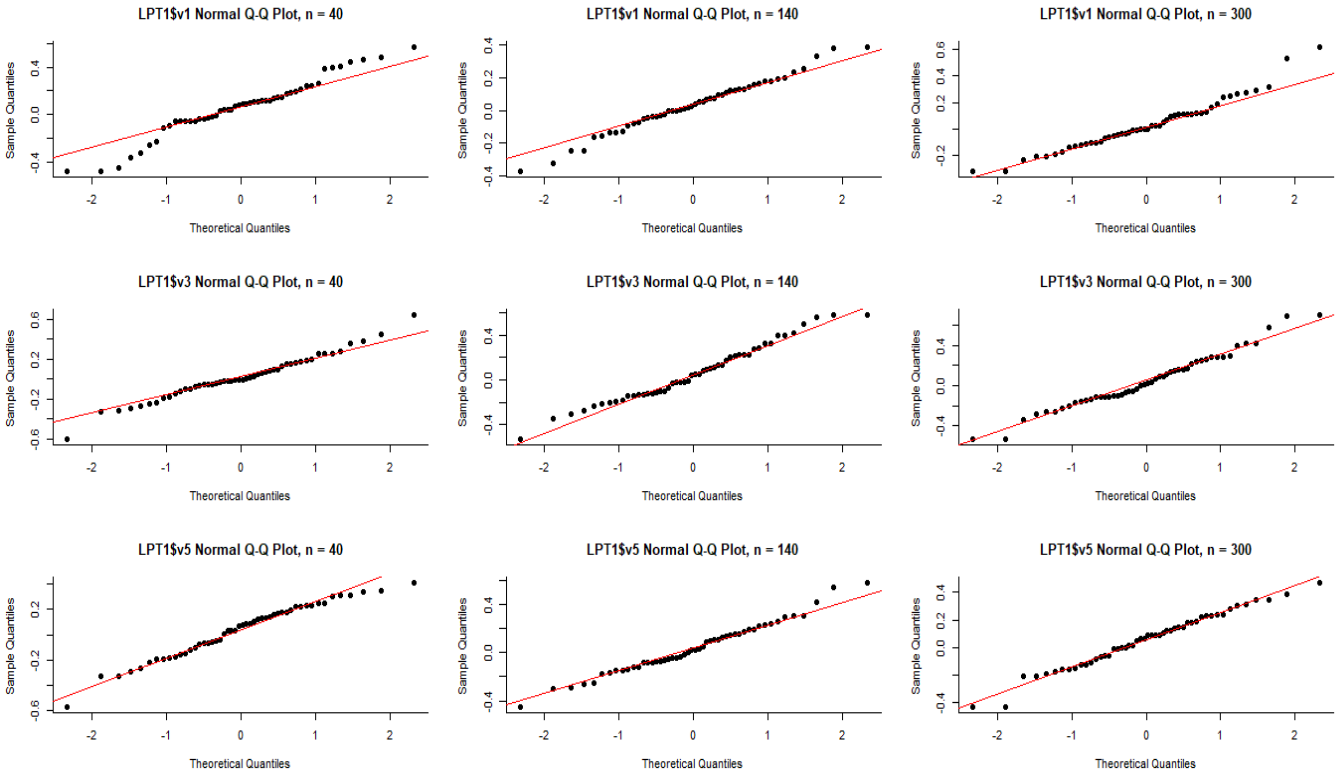

Fig. 3.8 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LPT} - \boldsymbol{\beta})$ *and* n *grows.*

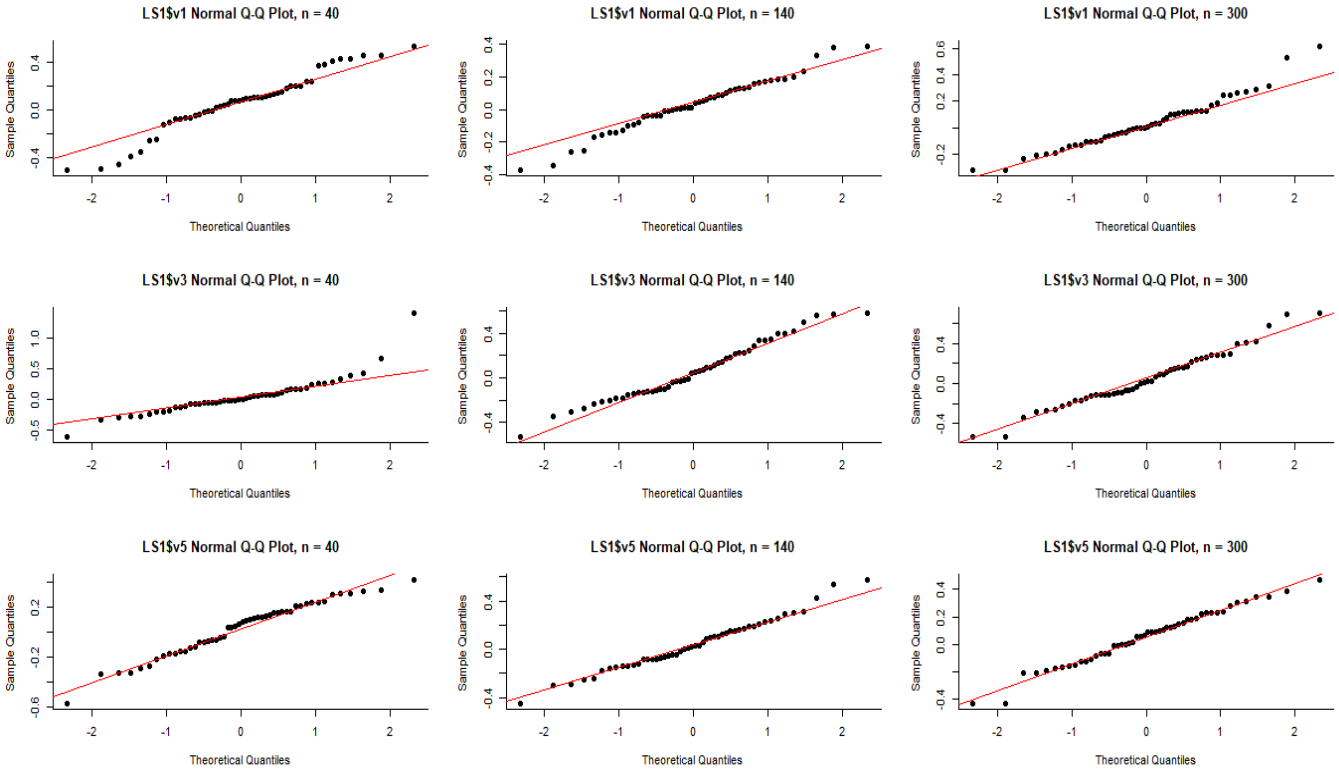

Fig. 3.9 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LS} - \boldsymbol{\beta})$ *and* n *grows.*

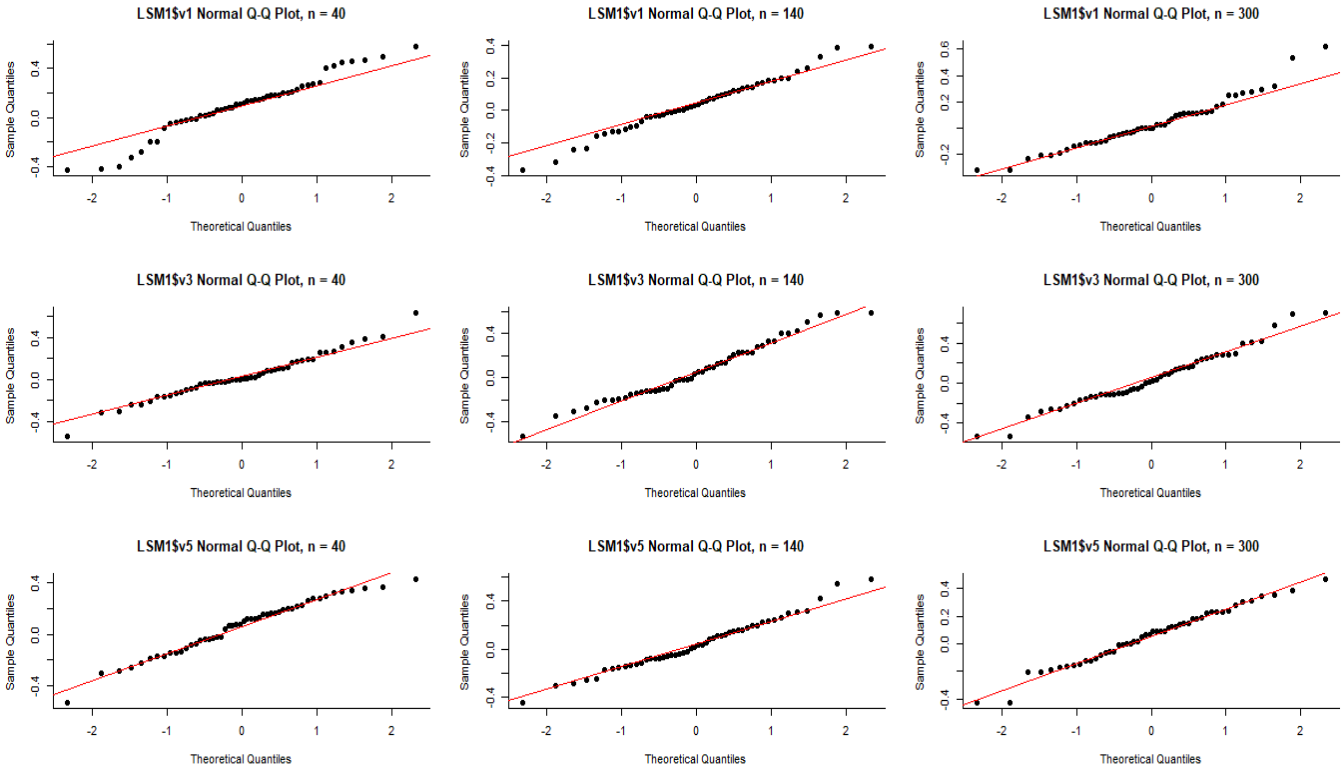

Fig. 3.10 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{LSM} - \boldsymbol{\beta})$ *and* n *grows.*

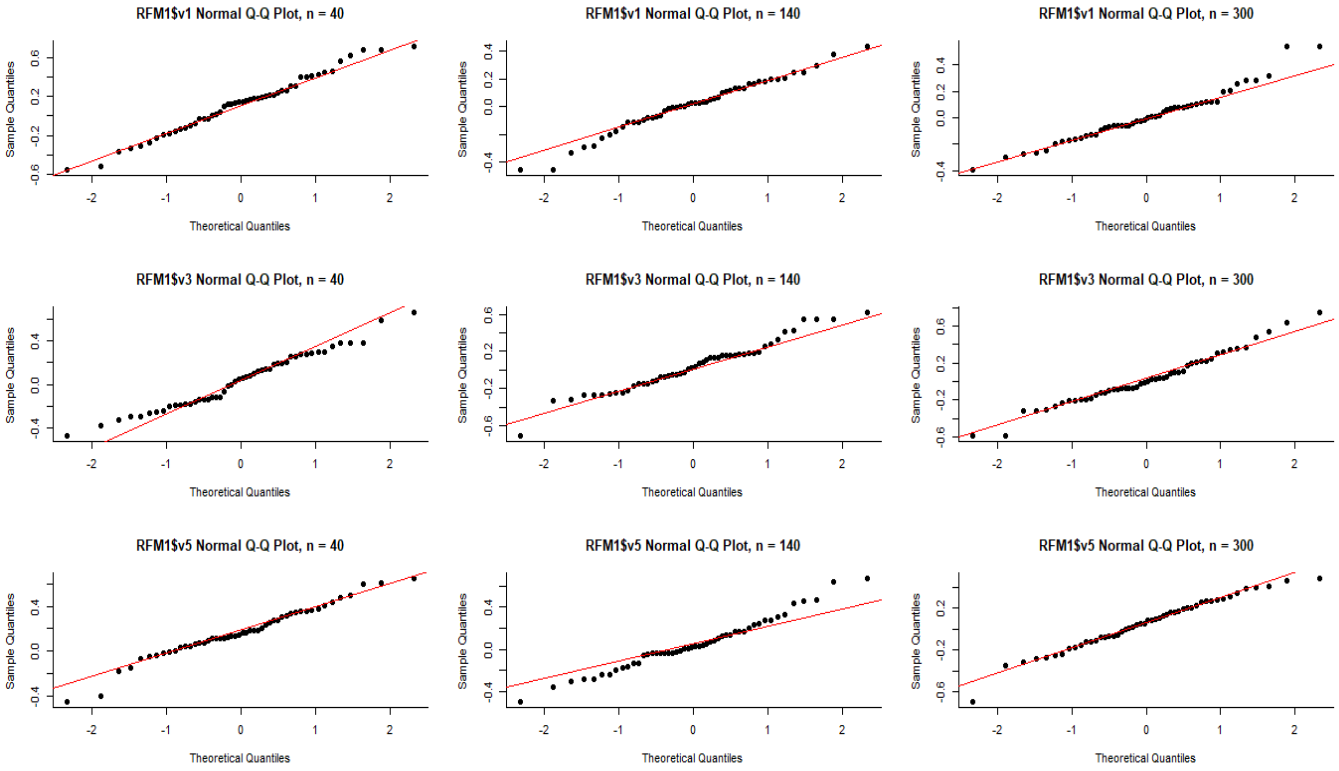

Fig. 3.11 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{RFM} - \boldsymbol{\beta})$ *and* n *grows.*

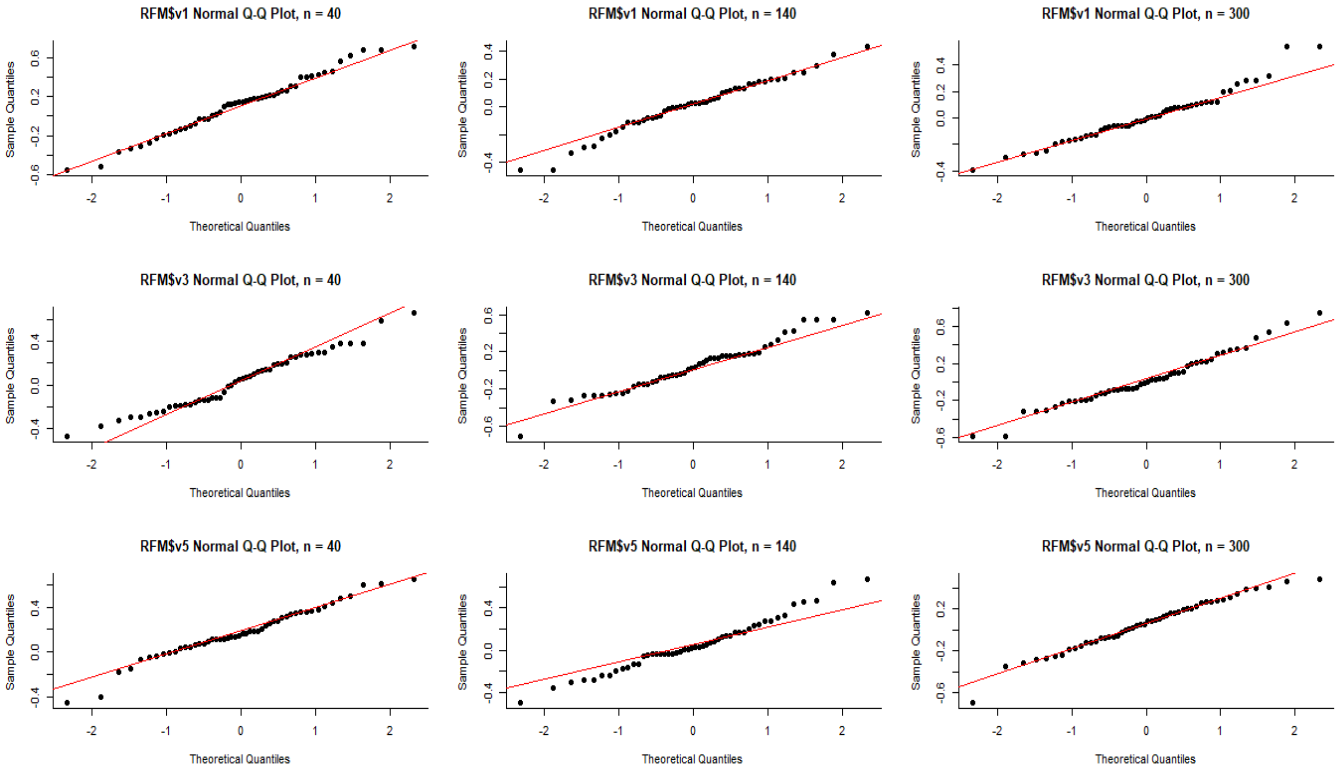

Fig. 3.12 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}^{RFM} - \boldsymbol{\beta})$ *and* n *grows.*

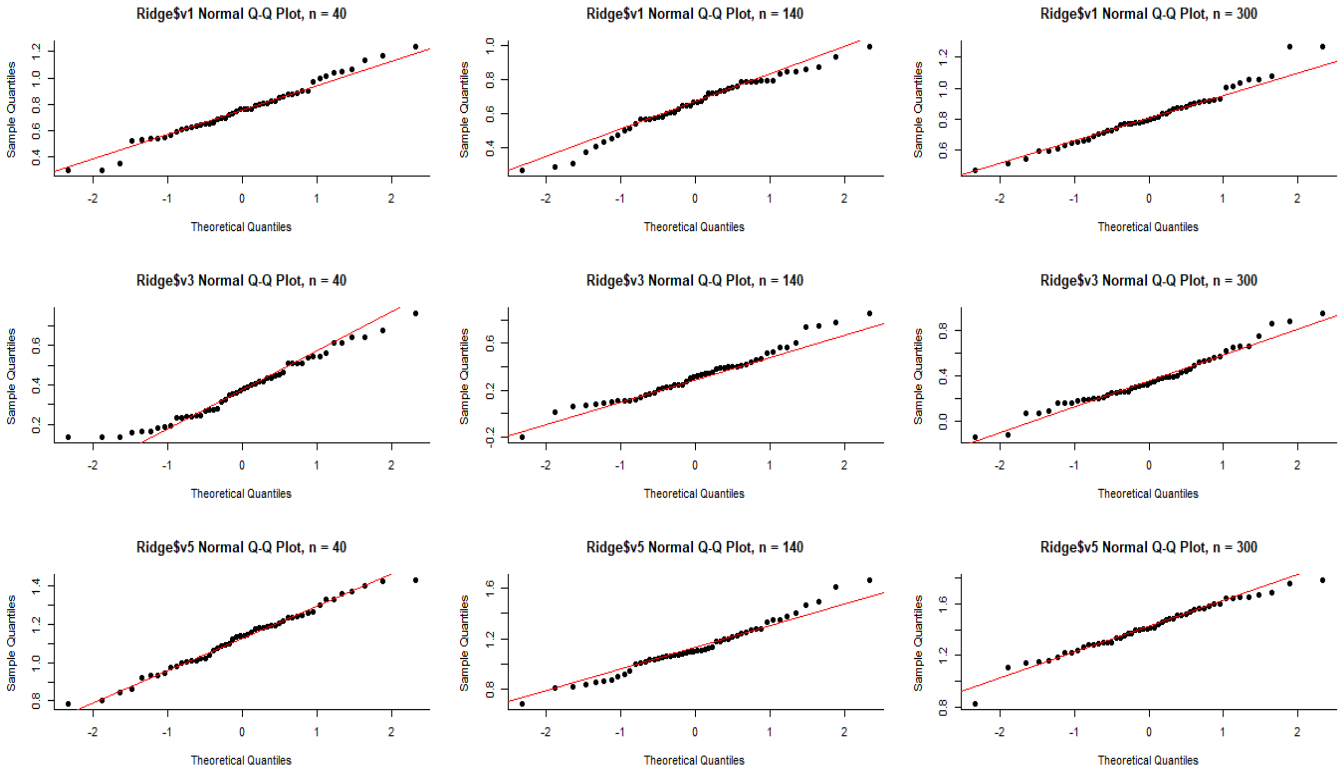

Fig. 3.13 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}^{Ridge} - \boldsymbol{\beta})$ *and* n *grows.*

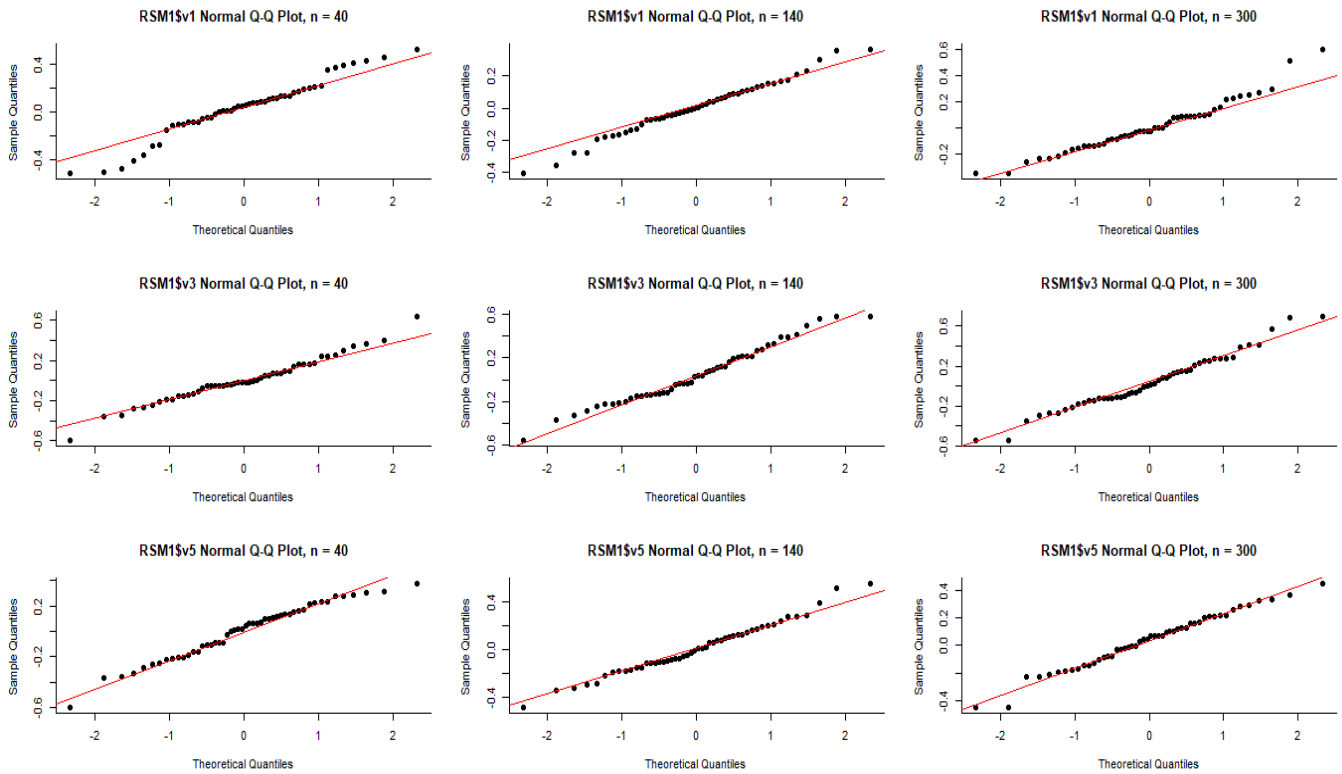

Fig. 3.14 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{n}(\widehat{\boldsymbol{\beta}}_1^{RSM} - \boldsymbol{\beta})$ *and* n *grows.*

Looking at the quantile-quantile plots above, we see the points match up along a straight line. While the line plotted is not a necessary component of the Q-Q plot, it allows us to visualize where the points should line up should the sample match the base distribution. Therefore from the plots above and according to the result (4.2), the aggregated estimators are approximately normal when n grows.

For asymptotic normality of $\mathbf{V} = (v_1, v_2, \ldots, v_p)^T = \sqrt{N}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})$, again three linear combinations are considered that extract v_1, v_3 and v_5 elements of the vector $\mathbf{V}$. Followings are the Q-Q plots to illustrate normality of v_1, v_3, and v_5 for $N = 2000$, $N = 7000$, and $N = 15000$ over 50 replications.

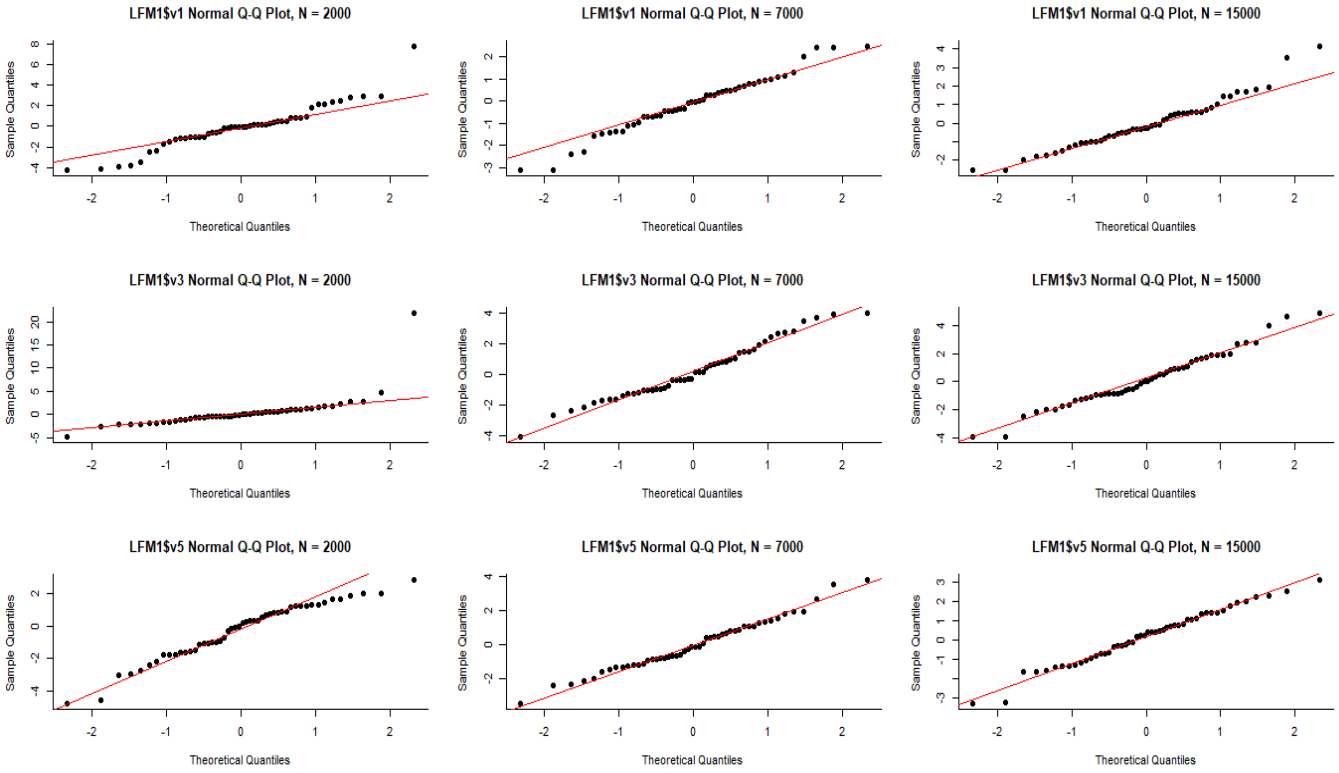

Fig. 3.15 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LFM} - \boldsymbol{\beta})$ *and* N *grows.*

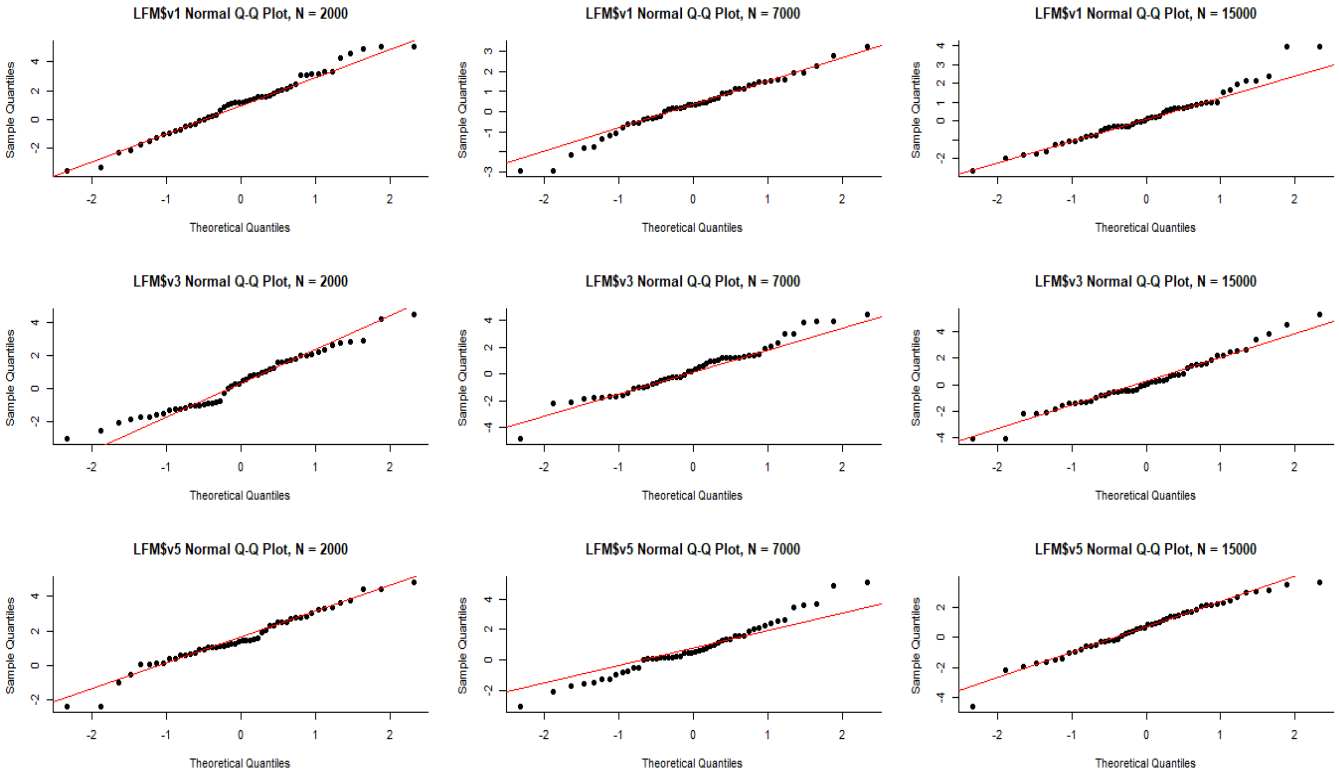

Fig. 3.16 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}^{LFM} - \boldsymbol{\beta})$ *and* N *grows.*

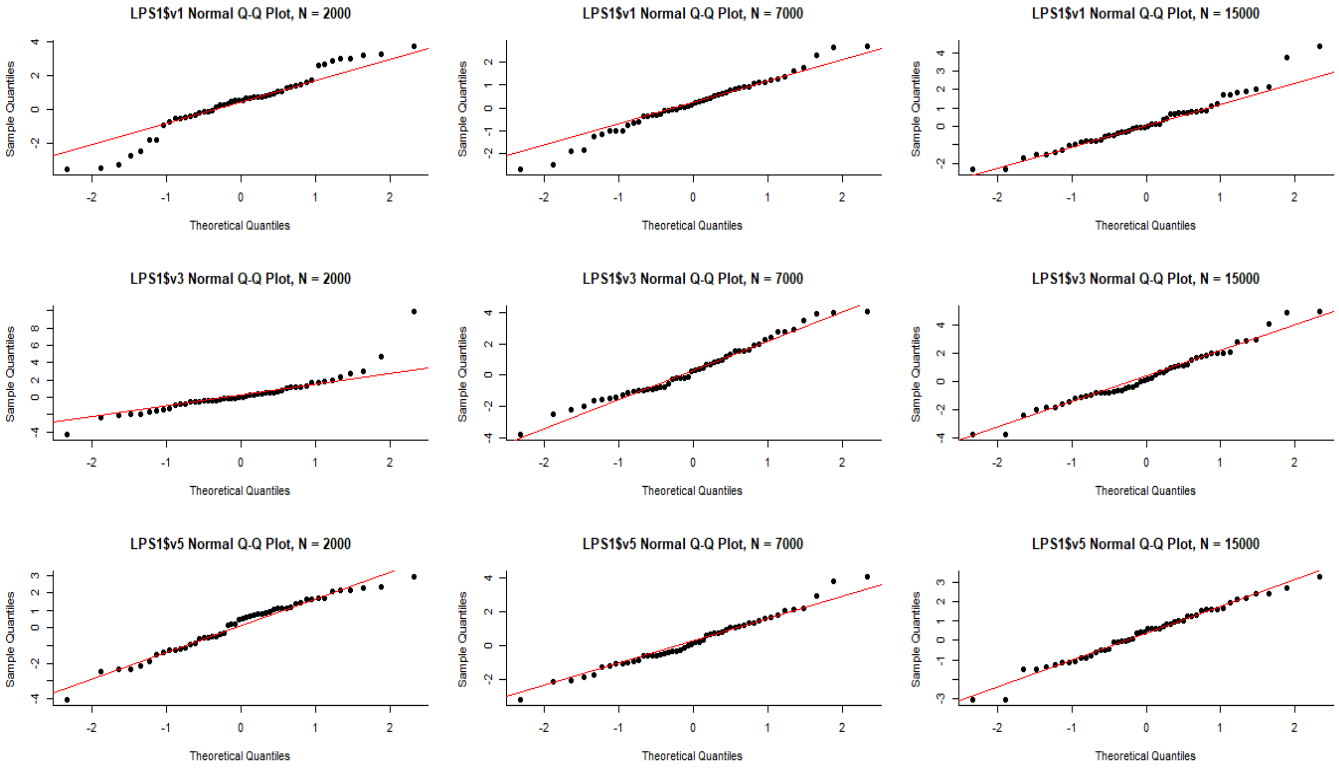

Fig. 3.17 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LPS} - \boldsymbol{\beta})$ *and* N *grows.*

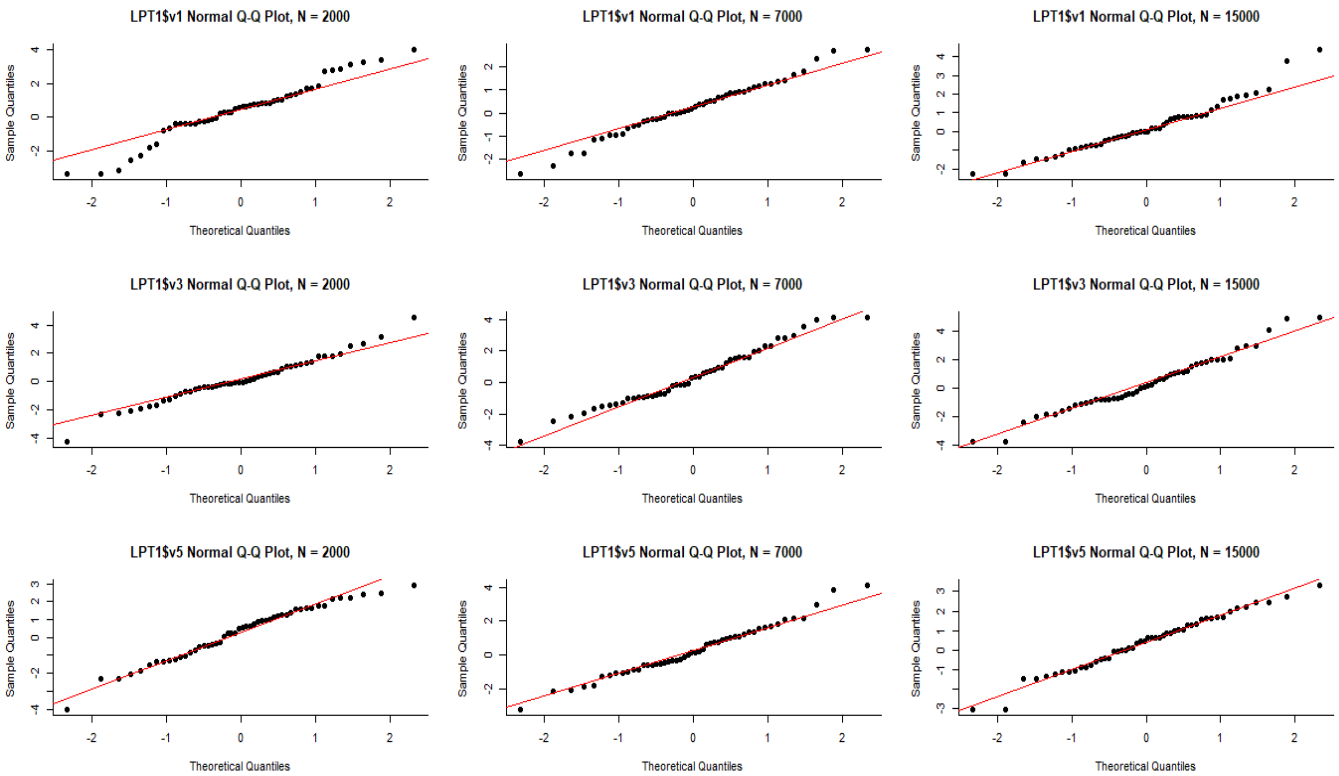

Fig. 3.18 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LPT} - \boldsymbol{\beta})$ *and* N *grows.*

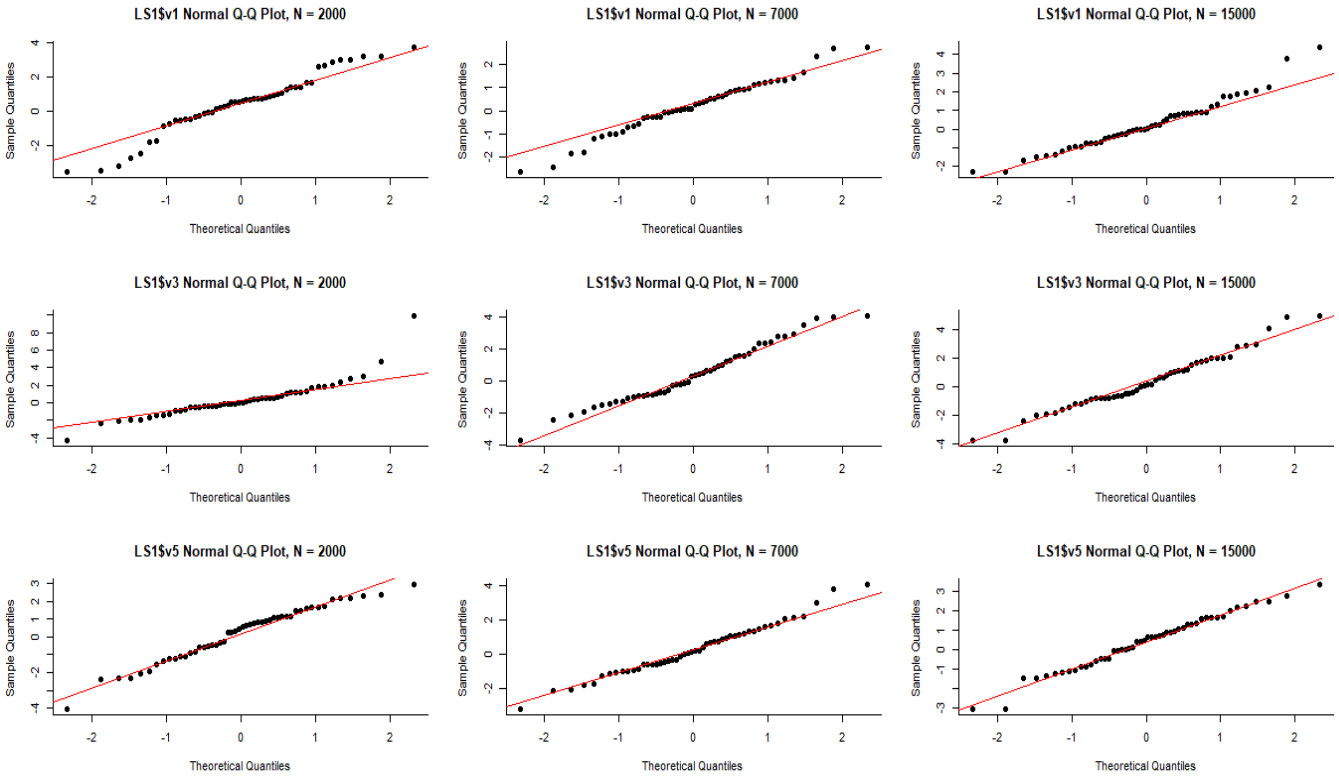

Fig. 3.19 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LS} - \boldsymbol{\beta})$ *and* N *grows.*

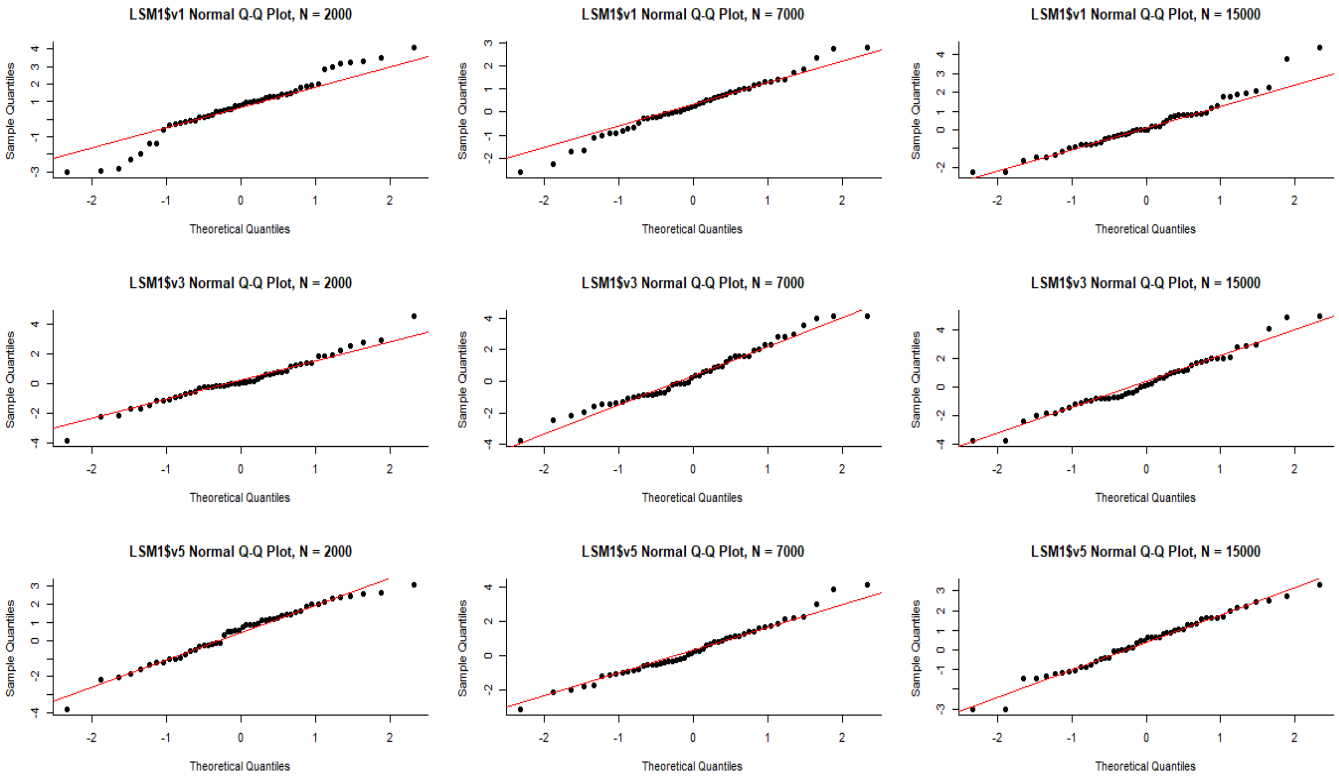

Fig. 3.20 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{LSM} - \boldsymbol{\beta})$ *and* N *grows.*

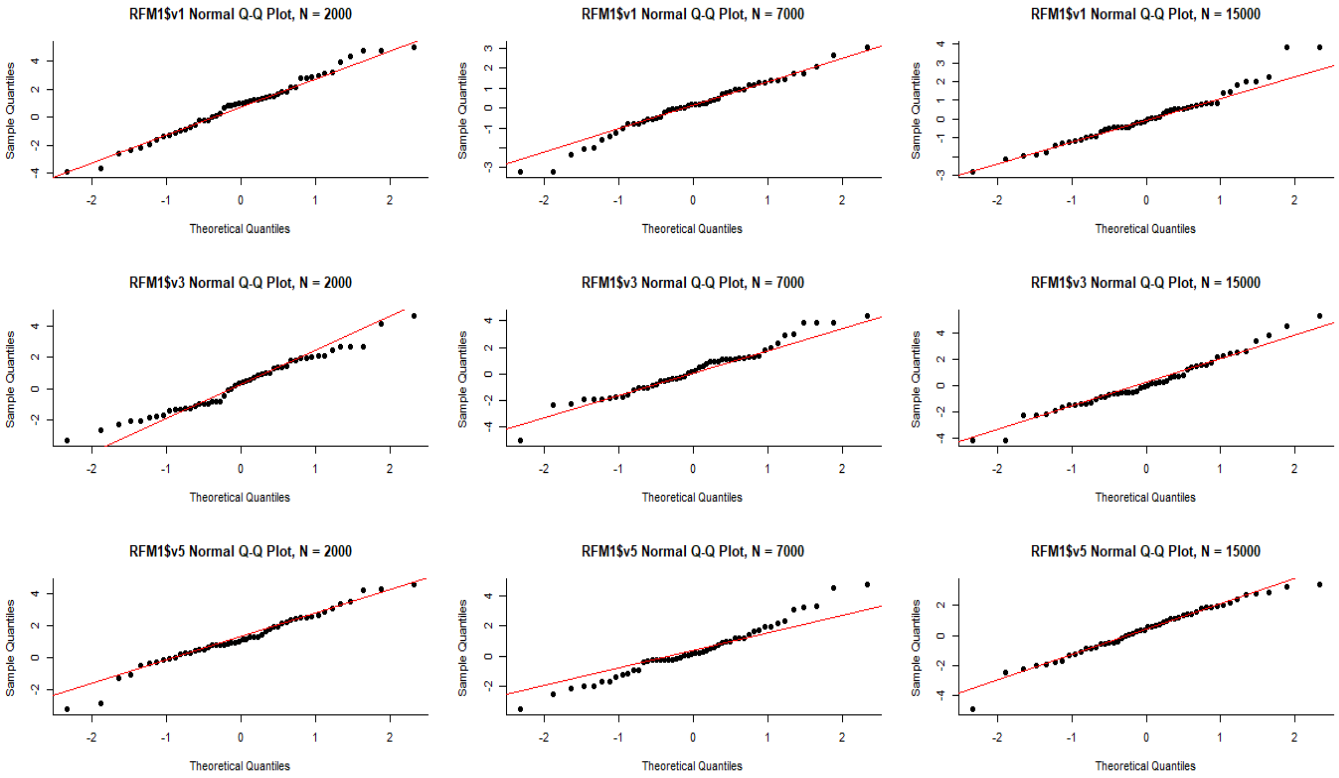

Fig. 3.21 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{RFM} - \boldsymbol{\beta})$ *and* N *grows.*

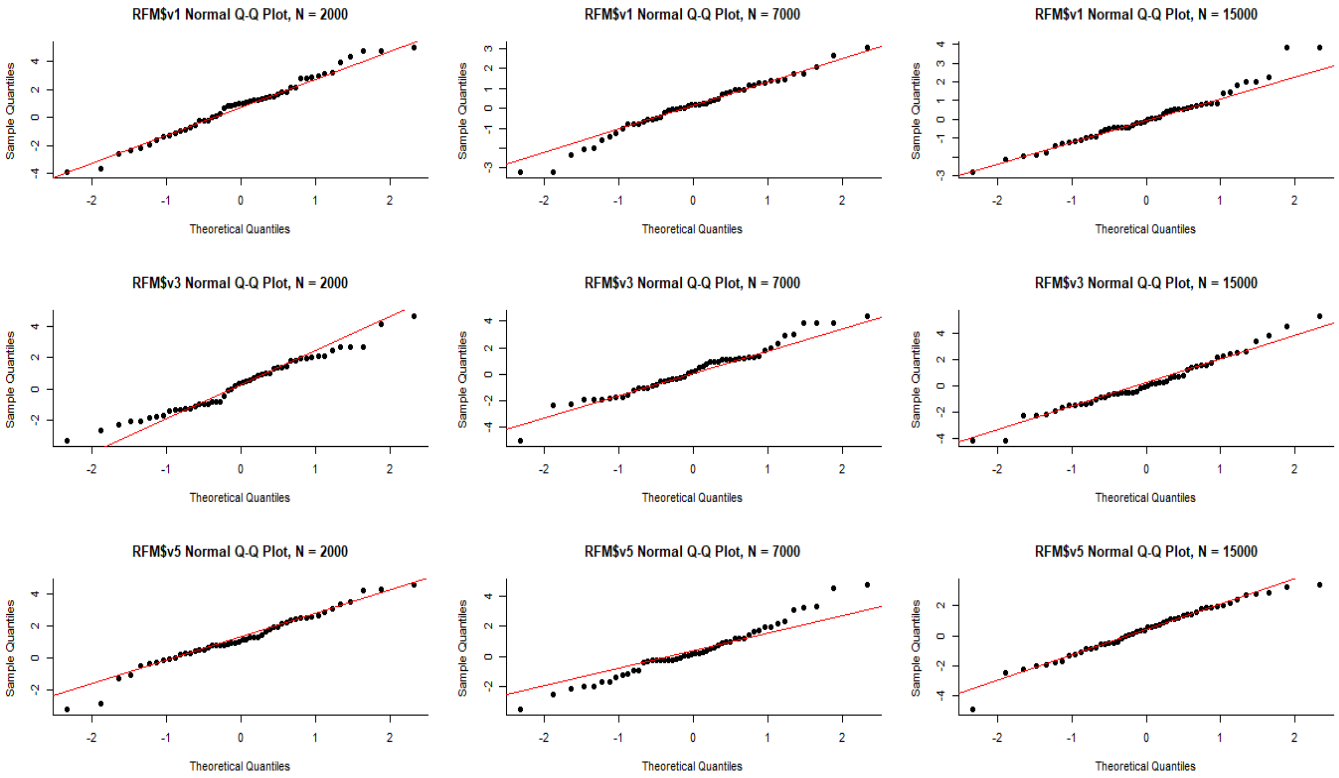

Fig. 3.22 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}^{RFM} - \boldsymbol{\beta})$ *and* N *grows.*

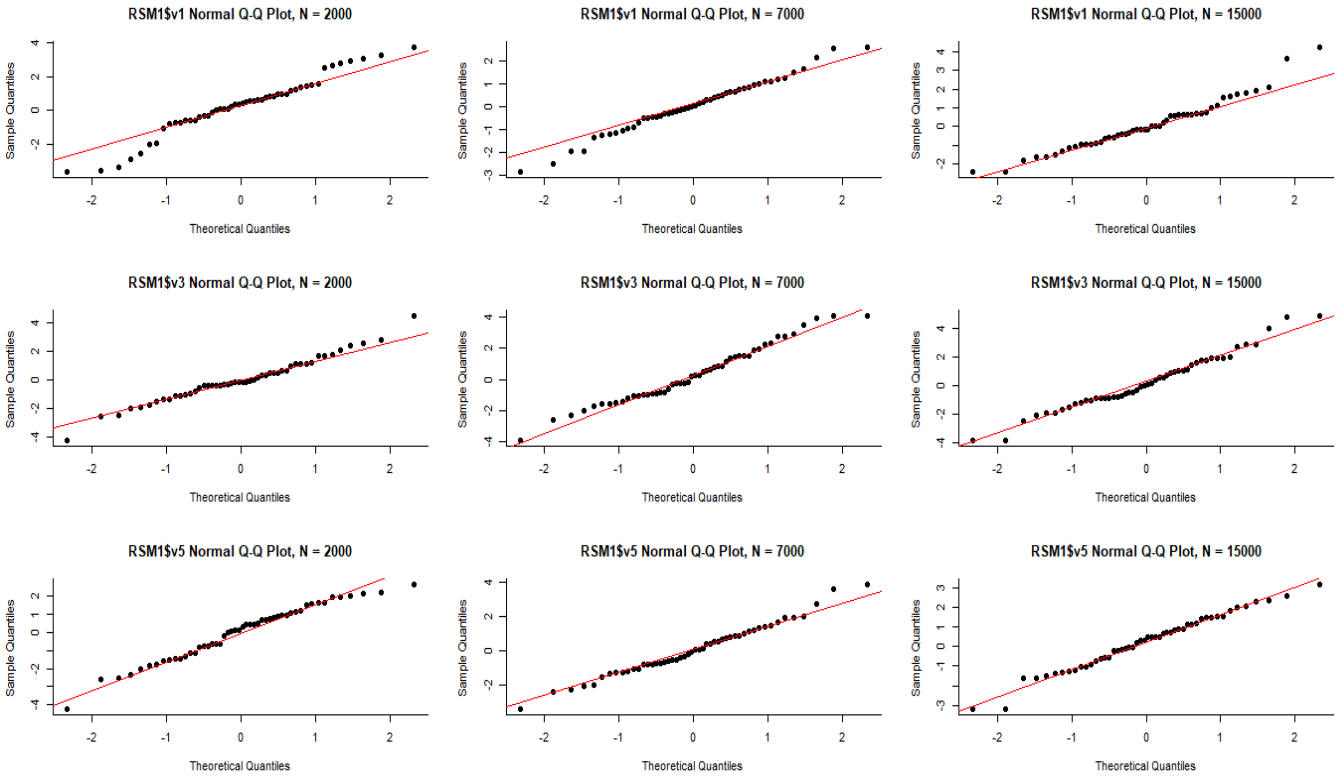

Fig. 3.23 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}_1^{RSM} - \boldsymbol{\beta})$ *and* N *grows.*

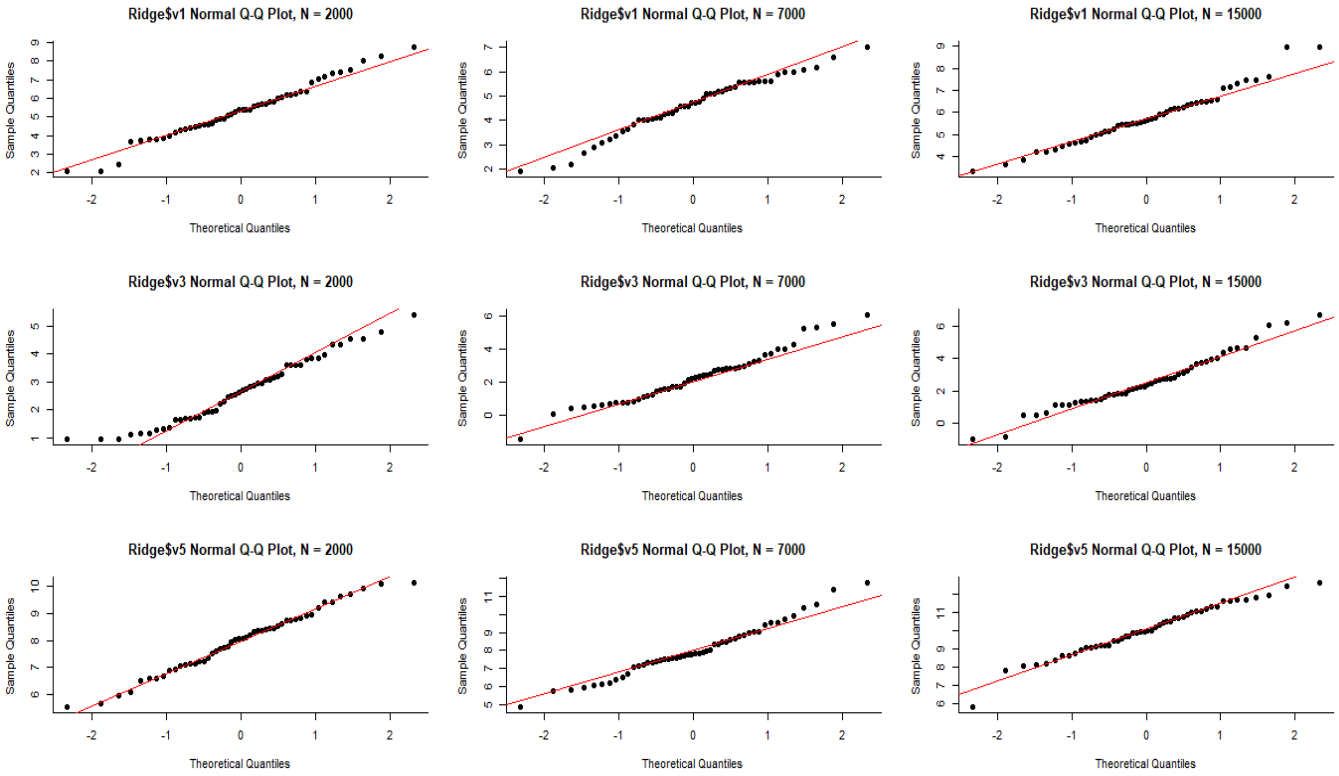

Fig. 3.24 *Normality of* v_1, v_2, v_3 *when* $\mathbf{V} = \sqrt{N}(\widehat{\boldsymbol{\beta}}^{Ridge} - \boldsymbol{\beta})$ *and* N *grows.*

Again, looking at the Q-Q plots for the aggregated estimators when N grows, we realize that the points are normally distributed.

Hence, through our simulation study in this section we illustrated that $\sqrt{n}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})$ and $\sqrt{N}(\widehat{\boldsymbol{\beta}}^{\blacktriangle} - \boldsymbol{\beta})$ are bounded and follow multivariate normal distribution when n and N grow. It thus provided our desired result which was $\sqrt{n}$-consistency and $\sqrt{N}$-consistency of the aggregated estimators where we used the new class of Liu-type shrinkage estimations proposed in Yüzbaşı et al. (2022).

3.4 Experiments on Empirical Data

In this section, we assess performance (the accuracy) of the discussed estimation strategies using an empirical data example. We have considered the Million Song Year Prediction Dataset (MSD) Bertin-Mahieux et al. (2011). The dataset is downloaded from UC Irvine Machine Learning Repository. It has $N = 515,345$ instances (the first $463,715$ for training and the last $51,630$ for testing) with $p = 90$ features. The MSD dataset was studied by Dobriban and Sheng (2019) to be analyzed in the framework of *distributed ridge regression*. Motivated by their work, we have considered the MSD dataset to assess performance of the distributed Liu-type shrinkage estimations. Similar to Dobriban and Sheng (2019), we try to predict the release year of a song from audio features. Figure (3.25), shows mean squared prediction error of the aggregated estimates on the MSD dataset.

We repeat the experiment 100 times, for each experiment, we randomly choose $N_{train} = 10,000$ samples from the training set and $N_{test} = 1,000$ from the test set. Next, we perform distributed Liu-type shrinkage estimations. As it was mentioned before, in application, in order to perform $\widehat{\boldsymbol{\beta}}_1^{RFM}$, $\widehat{\boldsymbol{\beta}}_1^{RSM}$, $\widehat{\boldsymbol{\beta}}_1^{LFM}$, $\widehat{\boldsymbol{\beta}}_1^{LSM}$, $\widehat{\boldsymbol{\beta}}_1^{LPT}$, $\widehat{\boldsymbol{\beta}}_1^{LS}$, and $\widehat{\boldsymbol{\beta}}_1^{LPS}$ estimation strategies we need to find significant and insignificant covariates. In this study, we have used AIC method, and it found that $p_1 = 78$ of covariates are significant and other $p_2 = 12$ are insignificant covariates. The number of local machines are chosen to be $K = 1, 10, 20, 50, 100$ that are displayed on the x axis of the plots.

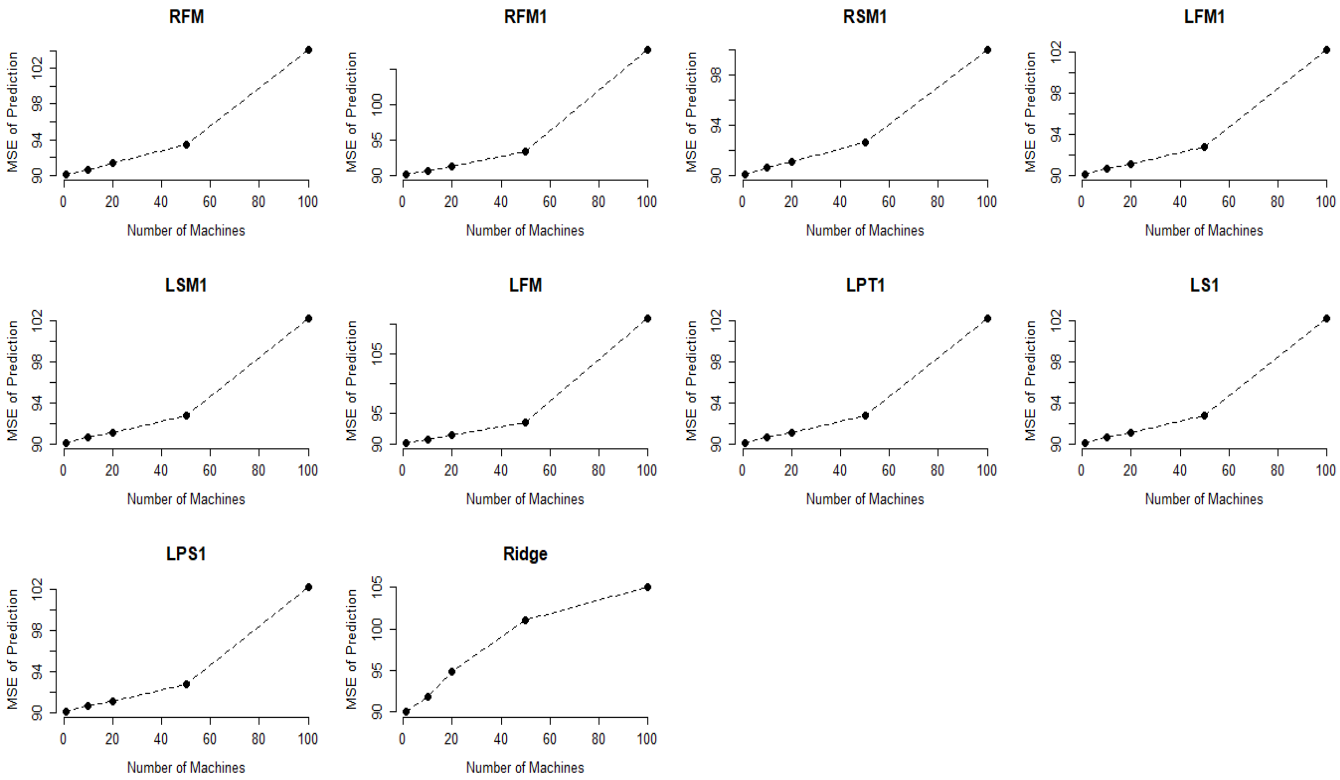

Fig. 3.25 *Million Song Year Prediction Dataset (MSD).*

It should be noted that, we have considered $K = 1$ to be in the experiment to compare the global estimation (i.e. $K = 1$) and the distributed estimation (i.e. $K = 10, 20, 50, 100$) methods. To do comparison between centralized (global estimator) and distributed setup (aggregated estimator), from the plots we can observe that,

1. Having a look at the plots, we see that prediction MSE of the discussed estimators grow nearly at the same rate and their performance are very similar together; Ridge estimator performs differently.

2. Comparing MSE of the aggregated estimators (or distributed estimation) with the global estimator tells us that, the distributed setting still performs relatively well while increasing the number of local machines.

3. The distributed Liu-type shrinkage estimators have smaller MSE's than distributed Ridge estimator even when we increase the number of machines.

To conclude, this example suggests a very positive outlook on using distributed Liu-type shrinkage estimation: The accuracy is affected very little even though the data is split up into 100 parts. Such datasets of large size cannot be analyzed on a single machine because there is always limits for memories. Moreover, we save at least 100x in computation time while we have nearly no loss in performance.

Chapter 4

Conclusion and future work

In this thesis, we have explored the idea of distributed supervised statistical learning and discussed the shrinkage methods proposed by Yüzbaşı et al. (2022) for distributed sparse linear regression analysis. Our objective was to assess the performance of these methods compare to the popular techniques such as ridge regression. After a review on distributed learning and the aggregation strategies, we investigated a new class of Liu-type shrinkage estimators proposed in Yüzbaşı et al. (2022). A simulation study was conducted to illustrate performance of the Liu-type shrinkage methods in the framework of distributed analysis. A real data example was investigated method, and also to applied the method. Through a simulation study, we have made several key findings that contribute to the understanding and advancement of distributed estimation. In fact, in this literature, we studied one-shot distributed Liu-type shrinkage in high dimensions.

First, we reviewed various aggregation strategies that are commonly used in distributed learning. We discussed the properties of averaging as an aggregation method and highlighted its advantages, such as simplicity and robustness against outliers. Furthermore, we discussed averaging method to be used for the new class of shrinkage methods proposed in Yüzbaşı et al. (2022). These shrinkage methods offer improved performance by combining the advantages of both variable selection and shrinkage. We examined the theoretical properties and algorithmic implementations of these shrinkage methods, highlighting their ability to handle high-dimensional datasets and provide more interpretable models in the framework of distributed learning.

To assess the performance of these shrinkage methods, we conducted a simulation study considering the mean squared error (MSE) criterion. Our results showed that the new class of shrinkage methods have comparable performance with ridge estimator and even outperform the ridge regression. This indicates that these methods offer better accuracy and predictive performance, particularly for high-dimensional data and sparse parameters settings in the framework of distributed regression analysis. These findings reinforce the importance of considering shrinkage methods as viable alternatives in distributed estimation tasks.

The simulation study demonstrated the superior performance of the aforementioned shrinkage methods compared to traditional techniques where the global estimation is used. It was addressed through this study that the number of machines to deploy is also another important question. However, according to the results from the simulation study, and also the properties we mentioned about the new class of shrinkage estimators it should be noted that these estimators are efficient even when the number of machines is provided by the nature of the problem of interest.

As it was mentioned before, tackling high dimension data needs high performance computing that is time and budget consuming. Despite the development in technology, it is still infeasible in practice to analyse such big data problems on a single machine in terms of memory capacity and computational power. Distributed analysis of big data problems is a bright idea that can be developed as a more advanced tool than its present form. The most important elements in the idea of distributed sparse linear regression are: first, using estimation strategies that can perform very close to their global version when they are used under the distributed setup, second, an optimal aggregation technique that is both practical and efficient in terms of accuracy. This thesis is an introductory work that has studied a specific class of estimation strategies to be used for distributed analysis.

As the field of distributed supervised statistical learning continues to evolve, it holds tremendous potential for various real-world applications, such as large-scale data analysis and predictive modeling. The findings from this thesis and the review work can be considered for further research and development, motivating the exploration of advanced distributed learning techniques and their application in diverse domains. In a future study, iterative distributed Liu-type shrinkage estimation can be performed which is a more consistent strategy as it was mentioned in the review sections of this thesis. Distributed post shrinkage strategy is also another interesting problem in the area of distributed learning. An important question is about the number of machines to deploy in a distributed setup, this is an optimization problem that still can be considered to be studied for different estimation strategies. Furthermore, it is of great importance to work on more aggregation strategies that can make the shrinkage methods more compatible with the distributed scheme and make the results even closer to the centralized estimation methods.

Bibliography

Achutegui, K., Crisan, D., Miguez, J., and Rios, G. (2014). A simple scheme for the parallelization of particle filters and its application to the tracking of complex stochastic systems. *ArXiv e-prints*.

Ahmed, S. E., Ahmed, F., and Yüzbaşı, B. (2023). *Post-Shrinkage Strategies in Statistical and Machine Learning for High Dimensional Data*. CRC Press.

Ai, M., Yu, J., Zhang, H., and Wang, H. (2021). Optimal subsampling algorithms for big data regressions. *Statistica Sinica*, 31(2):749–772.

Alheety, M. I. and Golam Kibria, B. (2013). Modified liu-type estimator based on (r-k) class estimator. *Communications in Statistics-Theory and Methods*, 42(2):304–319.

Barghi, H., Najafi, A., and Motahari, S. A. (2021). Distributed sparse feature selection in communication-restricted networks. *arXiv preprint arXiv:2111.02802*.

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *Annals of statistics*, 46(3):1352.

Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The million song dataset. 12th int. In *Conf. on Music Information Retrieval (ISMIR)*.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.

Chen, X., Liu, W., Mao, X., and Yang, Z. (2020). Distributed high-dimensional regression under a quantile loss function. *The Journal of Machine Learning Research*, 21(1):7432–7474.

Chen, X. and Xie, M.-g. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, pages 1655–1684.

Dobriban, E. and Sheng, Y. (2019). One-shot distributed ridge regression in high dimensions. *arXiv*, 2019.

Dobriban, E. and Sheng, Y. (2020). Wonder: weighted one-shot distributed ridgeregression in high dimensions. *The Journal of Machine Learning Research*, 21(1):2483–2534.

Dobriban, E. and Sheng, Y. (2021). Distributed linear regression by averaging. *The Annals of Statistics*, 49(2):918 – 943.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Zhang, Y. (2014). Optimality guarantees for distributed statistical estimation. *arXiv preprint arXiv:1405.0782*.

Fan, J., Guo, Y., and Wang, K. (2021). Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association*, pages 1–11.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fonseca, R. and Nadler, B. (2023). Distributed sparse linear regression under communication constraints. *arXiv preprint arXiv:2301.04022*.

Frank, L. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

Gao, Y., Liu, W., Wang, H., Wang, X., Yan, Y., and Zhang, R. (2022). A review of distributed statistical inference. *Statistical Theory and Related Fields*, 6(2):89–99.

Guestrin, C., Bodik, P., Thibaux, R., Paskin, M., and Madden, S. (2004). Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 1–10.

Hall, P. (2013). *The bootstrap and Edgeworth expansion*. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Huang, C. and Huo, X. (2019). A distributed one-step estimator. *Mathematical Programming*, 174:41–76.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909.

Johnson, R. A., Wichern, D. W., et al. (2002). *Applied multivariate statistical analysis*. Prentice hall Upper Saddle River, NJ.

Jordan, M. I., Lee, J. D., and Yang, Y. (2018). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681.

Kejian, L. (1993). A new class of blased estimate in linear regression. *Communications in Statistics-Theory and Methods*, 22(2):393–402.

Lee, J. D., Liu, Q., Sun, Y., and Taylor, J. E. (2017). Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1):115–144.

Liu, K. (2003). Using liu-type estimator to combat collinearity. *Communications in Statistics-Theory and Methods*, 32(5):1009–1020.

Liu, Q. and Ihler, A. T. (2014). Distributed estimation, information loss and exponential families. *Advances in neural information processing systems*, 27.

Ma, P., Mahoney, M., and Yu, B. (2014). A statistical perspective on algorithmic leveraging. In *International conference on machine learning*, pages 91–99. PMLR.

Mateos, G., Bazerque, J. A., and Giannakis, G. B. (2010). Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276.

McDonald, R., Hall, K., and Mann, G. (2010). Distributed training strategies for the structured perceptron. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 456–464.

Mcdonald, R., Mohri, M., Silberman, N., Walker, D., and Mann, G. (2009). Efficient large-scale distributed training of conditional maximum entropy models. *Advances in neural information processing systems*, 22.

McWilliams, B., Heinze, C., Meinshausen, N., Krummenacher, G., and Vanchinathan, H. P. (2014). Loco: Distributing ridge regression with random projections. *stat*, 1050:26.

Minsker, S. (2019). Distributed statistical estimation and rates of convergence in normal approximation. *Electronic Journal of Statistics*, 13:5213 – 5252.

Norouzirad, M. and Arashi, M. (2018). Preliminary test and stein-type shrinkage lasso-based estimators. *SORT-Statistics and Operations Research Transactions*, 42(1):45–58.

Politis, D. N., Romano, J. P., and Wolf, M. (1999). *Subsampling*. Springer Science & Business Media.

Predd, J. B., Kulkarni, S. B., and Poor, H. V. (2006). Distributed learning in wireless sensor networks. *IEEE signal processing magazine*, 23(4):56–69.

Rencher, A. C. and Schaalje, G. B. (2008). *Linear models in statistics*. John Wiley & Sons.

Rieder, H. (2012). *Robust Asymptotic Statistics: Volume I*. Springer Science & Business Media.

Rilstone, P., Srivastava, V. K., and Ullah, A. (1996). The second-order bias and mean squared error of nonlinear estimators. *Journal of Econometrics*, 75(2):369–395.

Rosenblatt, J. D. and Nadler, B. (2016). On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404.

Schifano, E. D., Wu, J., Wang, C., Yan, J., and Chen, M.-H. (2016). Online updating of statistical inference in the big data setting. *Technometrics*, 58(3):393–403.

Shalev-Shwartz, S. and Srebro, N. (2008). Svm optimization: inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning*, pages 928–935.

Shamir, O., Srebro, N., and Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. PMLR.

Shi, C., Lu, W., and Song, R. (2018). A massive data framework for m-estimators with cubic-rate. *Journal of the American Statistical Association*, 113(524):1698–1709.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Volgushev, S., Chao, S.-K., and Cheng, G. (2019). Distributed inference for quantile regression processes. *The Annals of Statistics*, 47(3):1634–1662.

Wang, C., Chen, M.-H., Wu, J., Yan, J., Zhang, Y., and Schifano, E. (2018a). Online updating method with new variables for big data streams. *Canadian Journal of Statistics*, 46(1):123–146.

Wang, H. (2019). More efficient estimation for logistic regression with optimal subsamples. *Journal of machine learning research*, 20:1–59.

Wang, H. and Ma, Y. (2021). Optimal subsampling for quantile regression in big data. *Biometrika*, 108(1):99–112.

Wang, H., Yang, M., and Stufken, J. (2019). Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525):393–405.

Wang, H., Zhu, R., and Ma, P. (2018b). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844.

Wang, J., Kolar, M., Srebro, N., and Zhang, T. (2017). Efficient distributed learning with sparsity. In *International conference on machine learning*, pages 3636–3645. PMLR.

Xue, Y., Wang, H., Yan, J., and Schifano, E. D. (2020). An online updating approach for testing the proportional hazards assumption with streams of survival data. *Biometrics*, 76(1):171–182.

Yüzbaşı, B., Ahmed, S. E., and Güngör, M. (2017). Improved penalty strategies in linear regression models. *REVSTAT-Statistical Journal*, 15(2):251–276.

Yüzbaşı, B., Asar, Y., and Ahmed, S. E. (2022). Liu-type shrinkage estimations in linear models. *Statistics*, 56(2):396–420.

Yüzbaşı, B. and Ejaz Ahmed, S. (2016). Shrinkage and penalized estimation in semi-parametric models with multicollinear data. *Journal of Statistical Computation and Simulation*, 86(17):3543–3561.

Zhang, C.-H. and Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576 – 593.

Zhang, H. and Wang, H. (2021). Distributed subdata selection for big data via sampling-based approach. *Computational Statistics & Data Analysis*, 153:107072.

Zhang, Y., Wainwright, M. J., and Duchi, J. C. (2012). Communication-efficient algorithms for statistical optimization. *Advances in neural information processing systems*, 25.

Zhao, T., Cheng, G., and Liu, H. (2016). A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400.

Zhu, X., Li, F., and Wang, H. (2021). Least-square approximation for a distributed system. *Journal of Computational and Graphical Statistics*, 30(4):1004–1018.

Zinkevich, M., Weimer, M., Li, L., and Smola, A. (2010). Parallelized stochastic gradient descent. *Advances in neural information processing systems*, 23.