mathematics

MDPI

Article

Efficient Estimation and Validation of Shrinkage Estimators in Big Data Analytics

Salomi du Plessis [1], Mohammad Arashi [1,2,*], Gaonyalelwe Maribe [1] and Salomon M. Millard [1]

[1] Department of Statistics, Faculty of Natural and Agricultural Science, University of Pretoria, Pretoria 0028, South Africa; u15176658@tuks.co.za (S.d.P.); g.maribe@up.ac.za (G.M.); sollie.millard@up.ac.za (S.M.M.)
[2] Department of Statistics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhdad, Mashhad 9177948974, Iran
* Correspondence: arashi@um.ac.ir

Abstract: Shrinkage estimators are often used to mitigate the consequences of multicollinearity in linear regression models. Despite the ease with which these techniques can be applied to small- or moderate-size datasets, they encounter significant challenges in the big data domain. Some of these challenges are that the volume of data often exceeds the storage capacity of a single computer and that the time required to obtain results becomes infeasible due to the computational burden of a high volume of data. We propose an algorithm for the efficient model estimation and validation of various well-known shrinkage estimators to be used in scenarios where the volume of the data is large. Our proposed algorithm utilises sufficient statistics that can be computed and updated at the row level, thus minimizing access to the entire dataset. A simulation study, as well as an application on a real-world dataset, illustrates the efficiency of the proposed approach.

Keywords: big data; efficient computation; Liu estimator; matrix of sufficient statistics; multicollinearity; ridge estimator

MSC: 62J07; 68T09

check for updates

Citation: du Plessis, S.; Arashi, M.; Maribe, G.; Millard, S.M. Efficient Estimation and Validation of Shrinkage Estimators in Big Data Analytics. *Mathematics* **2023**, *11*, 4632. https://doi.org/10.3390/math11224632

Academic Editor: Elvira Di Nardo

Received: 27 September 2023
Revised: 30 October 2023
Accepted: 8 November 2023
Published: 13 November 202312

1. Introduction

The ability to collect data for which the volume, velocity, and variety exceeds the capacity of standard analytical tools has been made possible by advancements in computer technology employed in practically every field of science and daily life. Access to these data types encourages innovation across a wide range of scientific fields but poses challenges to data storage, computational efficiency, and current statistical and computational methods [1]. Although the high variety and velocity aspects of big data require novel statistical methodology, the data considered in this paper speak to the high-volume aspect of big data. A dataset with a size exceeding 20% of a computer's RAM is considered large, whereas a dataset with a size exceeding 50% of the computer's RAM is considered massive since simple calculations would consume all remaining RAM [2]. Access to large datasets enables us to gain a better understanding of the relationship between a response variable and a set of predictor variables, but we often face major issues during the analysis of such data. Multicollinearity is one of these issues [3].

Multicollinearity refers to the existence of linear relationships between the predictor variables in a regression problem. Two major consequences of multicollinearity are unstable regression coefficients due to inflated standard errors and poor out-of-sample prediction performance. The consequences of multicollinearity, such as unstable regression coefficients, are often addressed with shrinkage estimators. Shrinkage estimators restrict the parameter space to avoid parameter explosion due to inflated standard errors. However, these estimators are often nonlinear with respect to the shrinkage parameter, which complicates their optimisation. For an overview of ongoing research relating to shrinkage estimators for multicollinearity and optimal tuning parameter selection, see [3,4].

Despite the ease with which these techniques can be applied to small- or moderate-size datasets, they encounter significant challenges in the big data domain. Some of these challenges are that the volume of the data frequently exceeds the storage capacity of a single computer and that the time required to obtain results becomes infeasible due to the computational burden of a high volume of data. Many strategies for the efficient computation of shrinkage estimators in ultrahigh dimensions exist, but the body of works relating to shrinkage estimators for datasets with many observations is limited. In the context of a large number of observations, Ref. [5] proposed an algorithm that utilises sufficient statistics to obtain the ridge estimator of the multiple linear regression model to address the above-mentioned big data challenges. Their proposal relied on a grid search strategy for finding the optimal shrinkage parameter.

In this paper, we extend the work of [5] by proposing a method for the efficient model estimation of various well-known ridge- and Liu type estimators with closed-from solutions for the optimal shrinkage parameter. Our approach also allows efficient model validation through *K*-fold cross-validation to evaluate the generalisability of the model. We utilise an array of sufficient statistics and can overcome the memory and computational efficiency barrier since we do not need access to the entire dataset simultaneously to obtain sufficient statistics. Furthermore, the dataset can be removed from memory once the sufficient statistic has been computed, and hence, minimal access to the dataset is required. For illustrative purposes, we consider five shrinkage estimators that aim to address multicollinearity in the linear regression model. These estimators are the ridge estimator [6], the modified ridge type estimator [7], the Liu estimator [8], the modified one-parameter Liu estimator [9] and the Kibria–Lukman estimator [10]. This paper is structured as follows: Section 2 introduces the linear regression model and the shrinkage estimators under consideration. An algorithm for efficient model estimation and validation is proposed in Section 3. Section 4 illustrates the efficiency of our proposed approach through a simulation study and application on a real-world dataset, and lastly, Section 5 concludes the paper.

2. Statistical Methodology

Consider the multiple linear regression model given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \tag{1}$$

with $\mathbf{y} = (y_1, \ldots, y_n)^\top$ the n-dimensional response variable, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ the $n \times p$ matrix of nonstochastic explanatory variables, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^\top$ the p-dimensional vector of regression coefficients, and $\mathbf{e} = (e_1, \ldots, e_n)^\top \sim N(0, \sigma^2\mathbf{I})$ the n-dimensional error vector. If rank($\mathbf{X}^\top\mathbf{X}$) = p, then let $\mathbf{Z} = \mathbf{XQ}$ and $\boldsymbol{\alpha} = \mathbf{Q}^\top\boldsymbol{\beta}$, where $\mathbf{Q}$ is the orthogonal matrix whose columns constitute the eigenvectors of $\mathbf{X}^\top\mathbf{X}$. Thus, the ordinary least squares estimator of $\boldsymbol{\alpha}$ is $\hat{\boldsymbol{\alpha}} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{y}$. Accordingly, $\hat{\sigma}^2 = \mathbf{y}^\top\left[\mathbf{I} - \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\right]\mathbf{y}/(n-p)$ and $V(\hat{\boldsymbol{\alpha}}) = \hat{\sigma}^2(\mathbf{Z}^\top\mathbf{Z})^{-1}$.

Now, consider any estimator of $\boldsymbol{\alpha}$ with the form

$$\hat{\boldsymbol{\alpha}}_g = g\left(\mathbf{Z}^\top\mathbf{Z}\right)\mathbf{Z}^\top\mathbf{y} \tag{2}$$

where $g \in \mathbf{R}^{n\times p} \to \mathbf{S}(p)$, with $\mathbf{S}(p)$ the space of all positive definite, symmetric, $p \times p$ matrices. The variance of $\hat{\boldsymbol{\alpha}}_g$ is given by

$$\begin{aligned} V(\hat{\boldsymbol{\alpha}}_g) &= g\left(\mathbf{Z}^\top\mathbf{Z}\right)\mathbf{Z}^\top COV(\mathbf{y})\mathbf{Z}g\left(\mathbf{Z}^\top\mathbf{Z}\right) \\ &= \sigma^2 g\left(\mathbf{Z}^\top\mathbf{Z}\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)g\left(\mathbf{Z}^\top\mathbf{Z}\right). \end{aligned}$$

In the presence of multicollinearity, $\mathbf{Z}^\top\mathbf{Z}$ is ill-conditioned, and as a result, the OLS estimator of $\boldsymbol{\alpha}$ has a large variance. Various shrinkage estimators have been proposed to mitigate the consequences of multicollinearity in the linear regression model. The shrinkage estimators considered in this paper are listed below. These can all be written in terms of the form given in (2), which is used in the algorithm proposed in Section 3.

1. The ridge estimator, with $k \geq 0$, proposed by [6] as a possible solution to multicollinearity is given by

$$\hat{\boldsymbol{\alpha}}_{RRE}(k) \quad = \quad \left(\mathbf{Z}^\top\mathbf{Z} + k\mathbf{I}_p\right)^{-1}\mathbf{Z}^\top\mathbf{y}.$$

   In this case, $g(\mathbf{Z}^\top\mathbf{Z}) = (\mathbf{Z}^\top\mathbf{Z} + k\mathbf{I}_p)^{-1}$. The shrinkage parameter, k, should be estimated using the data, but it is unclear which value of k produces the best estimator. A large number of efforts have been aimed at accurately estimating the shrinkage parameter. Some proposed techniques for estimating k were proposed by refs. [11–14] and recently, [15]. We consider the estimator of the shrinkage parameter proposed by [6]. The estimator is $\hat{k} = \hat{\sigma}^2/\hat{\alpha}^2_{max}$, where $\hat{\alpha}_{max}$ is the maximum element of $\hat{\boldsymbol{\alpha}}$.
2. The modified ridge-type estimator with $g(\mathbf{Z}^\top\mathbf{Z}) = (\mathbf{Z}^\top\mathbf{Z} + k(1+d)\mathbf{I}_p)^{-1}$ is

$$\hat{\boldsymbol{\alpha}}_{MRT}(k,d) \quad = \quad \left(\mathbf{Z}^\top\mathbf{Z} + k(1+d)\mathbf{I}_p\right)^{-1}\mathbf{Z}^\top\mathbf{y}.$$

   This estimator includes the ridge and OLS estimators as special cases. The authors of Ref. [7] suggest an iterative approach to estimating the shrinkage parameters. Their approach is as follows:
   (a) Let the initial estimate of d be calculated as $\hat{d} = min(\hat{\sigma}^2/\hat{\alpha}_i^2)$ where $\hat{\alpha}_i$ is the ith element of $\hat{\boldsymbol{\alpha}}$.
   (b) Using $\hat{d}$ from 1, estimate k as $\hat{k} = p\hat{\sigma}^2/\left(\sum_{i=1}^p\left(1+\hat{d}\right)\hat{\alpha}_i^2\right)$.
   (c) Let $\hat{d} = \hat{\sigma}^2/\left(k\hat{\alpha_i}^2\right) - 1$. Estimate d_{mrt} as $\hat{d}_{mrt} = p/\sum_{i=1}^p \hat{d}^{-1}$.
   (d) Use $\hat{d}_{mrt} = \hat{d}$ if $\hat{d}_{mrt}$ is not between 0 and 1.
3. The Liu estimator with the combined benefits of the ridge estimator and the Stein type estimator [16] was proposed by [8]. Given that rank($\mathbf{X}^\top\mathbf{X}$) = p, the Liu estimator is given by

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_{LIU}(d) \quad &= \quad \left(\mathbf{Z}^\top\mathbf{Z} + \mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{y} + d\hat{\boldsymbol{\alpha}}\right) \\ &= \quad \left(\mathbf{Z}^\top\mathbf{Z} + \mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{Z} + d\mathbf{I}_p\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\mathbf{y}\end{aligned}$$

   with $g(\mathbf{Z}^\top\mathbf{Z}) = (\mathbf{Z}^\top\mathbf{Z} + \mathbf{I}_p)^{-1}(\mathbf{Z}^\top\mathbf{Z} + d\mathbf{I}_p)(\mathbf{Z}^\top\mathbf{Z})^{-1}$. The estimator of the shrinkage parameter is

$$\hat{d} \quad = \quad 1 - \hat{\sigma}^2\left[\frac{\sum_{i=1}^p(1/(\lambda_i(\lambda_i+1)))}{\sum_{i=1}^p\left(\hat{\alpha_i}^2/(\lambda_i+1)^2\right)}\right]$$

   where $\hat{\alpha}_i$ and λ_i are the ith element of $\hat{\boldsymbol{\alpha}}$ and the vector of eigenvalues of $\mathbf{X}^\top\mathbf{X}$, respectively.
4. A limitation of the estimator of the shrinkage parameter of the Liu estimator is that, in some instances, it has a negative value that affects the estimator's performance [17]. The modified one-parameter Liu estimator proposed by [9] yields a positive value of $\hat{d}$ and provides a significant improvement in the performance of the estimator. Given that rank($\mathbf{X}^\top\mathbf{X}$) = p, the modified one-parameter Liu estimator is given by

$$\hat{\boldsymbol{\alpha}}_{MLIU}(d) \quad = \quad \left(\mathbf{Z}^\top\mathbf{Z} + \mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{Z} - d\mathbf{I}_p\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\mathbf{y}$$

with $g(\mathbf{Z}^\top\mathbf{Z}) = \left(\mathbf{Z}^\top\mathbf{Z}+\mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{Z}-d\mathbf{I}_p\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}$. The optimal value of the shrinkage parameter is

$$\hat{d} \;=\; \min\left(\frac{\lambda_i(\hat{\sigma}^2+\hat{\alpha}_i{}^2)}{\hat{\sigma}^2+\lambda_i\hat{\alpha}_i{}^2}\right)$$

where $\hat{\alpha}_i$ and λ_i are the ith element of $\hat{\boldsymbol{\alpha}}$ and the vector of eigenvalues of $\mathbf{X}^\top\mathbf{X}$, respectively.

5. Lastly, the Kibria–Lukman estimator, proposed by [10], is a one-parameter estimator that combines the characteristics of both the ridge and Liu estimators. Given that rank($\mathbf{X}^\top\mathbf{X}$) = p, the estimator is given by

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_{KL}(k) &= \left(\mathbf{Z}^\top\mathbf{Z}+k\mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{y}-k\hat{\boldsymbol{\alpha}}\right)\\ &= \left(\mathbf{Z}^\top\mathbf{Z}+k\mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{Z}-k\mathbf{I}_p\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\mathbf{y}\end{aligned}$$

with $g(\mathbf{Z}^\top\mathbf{Z}) = \left(\mathbf{Z}^\top\mathbf{Z}+k\mathbf{I}_p\right)^{-1}\left(\mathbf{Z}^\top\mathbf{Z}-k\mathbf{I}_p\right)\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}$. The estimator of the shrinkage parameter proposed by [10], is

$$\hat{k} \;=\; \frac{\hat{\sigma}^2}{2\hat{\alpha}_i{}^2+(\hat{\sigma}^2/\lambda_i)}$$

with $\hat{\alpha}_i$ the ith element of $\hat{\boldsymbol{\alpha}}$ and λ_i the ith eigenvalue of $\mathbf{X}^\top\mathbf{X}$. Following [18], the harmonic mean version is given by

$$\hat{k} \;=\; \frac{\hat{\sigma}^2}{\sum_{i=1}^{p}\left[2\hat{\alpha}_i{}^2+(\hat{\sigma}^2/\lambda_i)\right]}.$$

3. Efficient Model Estimation and Validation in Big Data Analytics

The consequences of multicollinearity in linear regression models are often mitigated using shrinkage methods. Despite the ease with which these techniques can be applied to small or moderate datasets, they encounter significant challenges in the big data domain. Some of these challenges are that the high volume of data frequently exceeds the storage capacity of a single computer and that the time required to obtain results becomes infeasible due to the computational burden of such a high volume of data. To address these big data challenges, Ref. [5] proposed an algorithm that utilises an array of sufficient statistics to obtain the ridge estimator of the multiple linear regression model. The sufficient statistics array can be computed and updated at the row or mini-batch level. However, the ridge estimator is nonlinear with respect to the shrinkage parameter, which complicates its optimisation. The shrinkage estimators given in Section 2 are frequently utilised to address the adverse effects of multicollinearity in linear regression models and has the added advantage of having closed-form solutions. In this section, we extend the work of [5] to include various other well-known shrinkage estimators and also adapt their algorithm to allow an efficient model validation by employing K-fold cross-validation. We utilise the array of sufficient statistics given in (3) in our proposed approach. The sufficient statistic array is used to determine the estimator of the regression coefficients as well as the estimator of the covariance matrix of the regression coefficients in the multiple linear regression model when using the estimators in Section 2. For estimation purposes, access to the entire dataset is not needed once the sufficient statistics array has been computed. The sufficient statistics array for shrinkage estimators with the form given in (2) is

$$\mathbf{A} := \sum_{i=1}^{n}\mathbf{A}_i = \begin{bmatrix} S_{yy} & \mathbf{S}_{zy}^\top \\ \mathbf{S}_{zy} & \mathbf{S}_{zz}\end{bmatrix} = \begin{bmatrix}\sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i\mathbf{z}_i^\top \\ \sum_{i=1}^{n}\mathbf{z}_i y_i & \sum_{i=1}^{n}\mathbf{z}_i\mathbf{z}_i^\top\end{bmatrix}. \tag{3}$$

Considering the sufficient statistics array, the form of the estimators in Section 2 is

$$\hat{\boldsymbol{\alpha}}_g = g(\mathbf{S}_{zz})\mathbf{S}_{zy}.$$

The storage capacity of a single computer may be much less than the size of a large dataset, and hence the computation of (3) has to be distributed across multiple processors, in which case, algorithms based on multiple processors should be utilised. Parallel computation is extremely useful in this regard, and hence the concept of MapReduce is important in our approach since it interleaves parallel and sequential computation. The MapReduce concept enables us to perform K-fold cross-validation with ease, and hence we base our algorithm on a strategy suitable for multiple processors. Model validation often relies on resampling techniques that utilise two datasets, one for training the model and one for evaluating the prediction performance of the model on unseen data. The testing dataset must be accessed after the regression coefficients are estimated, irrespective of the method used to obtain the regression coefficients, since the predicted response is calculated using only the observed testing data and the estimated regression coefficients. The advantage of using K-fold cross-validation as a model validation technique lies in the fact that (3) can be obtained for each of the K training datasets without any more access to the data than would be required to calculate (3) for the complete dataset. Suppose we partition the dataset into K blocks containing n_k observations. The sufficient statistics array of each block is then given by $\sum_{i=1}^{n_k} \mathbf{A}_i$, such that the sufficient statistics array of the complete dataset is

$$\mathbf{A} := \sum_{k=1}^{K} \mathbf{A}_{(k)} := \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathbf{A}_i.$$

We can easily obtain the array of sufficient statistics for any training dataset since each of the K training datasets contains $(K-1)$ of the K blocks. Algorithm 1 is used for the model estimation and validation of the shrinkage estimators given in Section 2. Furthermore, Algorithm 1 is partitioned into four stages. The eigenvectors and eigenvalues of $\mathbf{X}^\top\mathbf{X}$ corresponding to the K training datasets are calculated in the first stage. The second step involves calculating the sufficient statistics array for each of the K blocks. In the third stage, we obtain the estimators proposed in Section 2, and finally, the prediction performance is evaluated in the fourth stage.

It is important to note that the estimators we considered have a general mathematical form, and hence the approach is not limited to only those estimators presented in this paper. Should feature selection be of interest, our approach can be applied to various other penalties, such as the elastic net, by using the local quadratic approximation of [19]. In these cases, it is not necessary to use the canonical form of the multiple linear regression model since the optimal turning parameter needs to be obtained through K-fold cross-validation. Furthermore, with minor modifications to the algorithm, data issues such as outliers could be addressed with the robust ridge estimator [20]. However, our main concern is multicollinearity, and hence the simulation and application considered the five estimators given in Section 2.

Algorithm 1 Model estimation and validation based on the sufficient statistics array

```
Input: Subsets of the data
Output: $\hat{\boldsymbol{\alpha}}_g$, $\hat{\sigma}^2$, $V\left(\hat{\boldsymbol{\alpha}}_g\right)$ and measures of prediction performance based on the testing dataset.
1. Divide the data into $K$ blocks.
MAP tasks
2. For each of the $K$ blocks:
    2.1 Calculate $\mathbf{S}_{xx(k)} = \sum_{i=1}^{n_k} \mathbf{x}_i \mathbf{x}_i^\top$
Reduce tasks
3. Let $j$ be the index of the block to be excluded from each of the $K$ training datasets such that $-j$ represent the remaining $K-1$ blocks.
For the $K$ training datasets:
    3.1 Calculate $\sum_{k \neq j} \mathbf{S}_{xx(k)}$
    3.2 Calculate $\mathbf{Q}_{(-j)}$, the eigenvectors of $\sum_{k \neq j} \mathbf{S}_{xx(k)}$.
    3.3 Calculate $\left(\lambda_1, \ldots, \lambda_p\right)_{(-j)}$, the eigenvalues of $\sum_{k \neq j} \mathbf{S}_{xx(k)}$.
MAP tasks
4. For each of the $K$ blocks:
    4.1 Calculate $S_{yy(k)}$, $\mathbf{S}_{zy(k)}$, $\mathbf{S}_{zz(k)}$ and $\mathbf{A}_{(k)}$ for each $\mathbf{Q}_{(-j)}$ with $j \in \{1, \ldots, K\}$.
Reduce tasks
5. For each of the $K$ training datasets with $j \in \{1, \ldots, K\}$:
    5.1 Calculate $\sum_{k \neq j} \mathbf{A}_{(k)}$.
    5.2 Calculate $\hat{\boldsymbol{\alpha}}_{(-j)}$, and $\hat{\sigma}^2_{(-j)}$.
    5.3 Calculate the shrinkage parameter of the estimator under consideration.
    5.4 Calculate $\hat{\boldsymbol{\alpha}}_g$ and $V\left(\hat{\boldsymbol{\alpha}}_g\right)$.
6. For each of the $K$ testing datasets:
    6.1 Calculate the predicted response.
    6.2 Calculate prediction performance measures based on the observed and predicted responses.
```

4. Numerical Analysis

In this section, the performance of the proposed approach is evaluated through an extensive simulation study and an application on a real-world dataset. A high-performance computer with 60 cores and 256GB RAM was used for all computations.

4.1. Simulation Study

The simulation study considered two experiments. The first experiment aimed to compare the execution time of our proposed approach with the execution time of the traditional approach, where we performed the relevant matrix algebra from Section 2 using the entire dataset at once. The second experiment was to verify that our algorithm yielded the same estimators and performance measures as the traditional approach. Theoretically, our proposed approach should yield the same answers as the traditional approach, which was verified by the results of the second experiment. The predictor variables were generated using

$$X_{ij} = \left(1 - \gamma^2\right)^{1/2} z_{ij} + \gamma z_{i(p+1)}$$

with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, p\}$, z_{ij} independent standard normal random numbers and γ^2 the correlation between any two predictor variables. The response was generated by

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + e_i$$

with $i \in \{1, \ldots, n\}$ and $e_i \sim N\left(0, \sigma^2\right)$. We considered $n = 10{,}000{,}000$, $p = 100$, $\gamma = 0.9, 0.99$, and $\sigma^2 = 1$. The regression coefficients were generated from a uniform$(-1, 1)$ distribution. Furthermore, we included both 5-fold and 10-fold cross-validation and considered three performance measures, namely the mean squared error (MSE) of the estimators, the mean prediction error (MPE) of the predicted response on the testing dataset, and the symmetric mean absolute percentage error (SMAPE) of the predicted response on the testing dataset. For an extended performance evaluation, the measures used by [21] can also be considered. The MSE, MPE, and SMAPE were calculated using

$$\begin{aligned}
\mathrm{MSE}(\hat{\boldsymbol{\beta}}) &= \frac{1}{K}\sum_{k=1}^{K}\left(\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}\right)^{\top}\left(\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}\right) \\
\mathrm{MPE}(\hat{\boldsymbol{y}}) &= \frac{1}{n}\sum_{k=1}^{K}(\hat{\boldsymbol{y}}_k - \boldsymbol{y})^{\top}(\hat{\boldsymbol{y}}_k - \boldsymbol{y}). \\
\mathrm{SMAPE}(\hat{\boldsymbol{y}}) &= \frac{100\%}{n}\sum_{k=1}^{K}\sum_{i=1}^{\frac{n}{K}}\frac{|\hat{y}_{ik} - y_{ik}|}{(|y_{ik}| + |\hat{y}_{ik}|)/2}.
\end{aligned}$$

Figure 1 and Table 1 contain the results obtained in the simulation study.

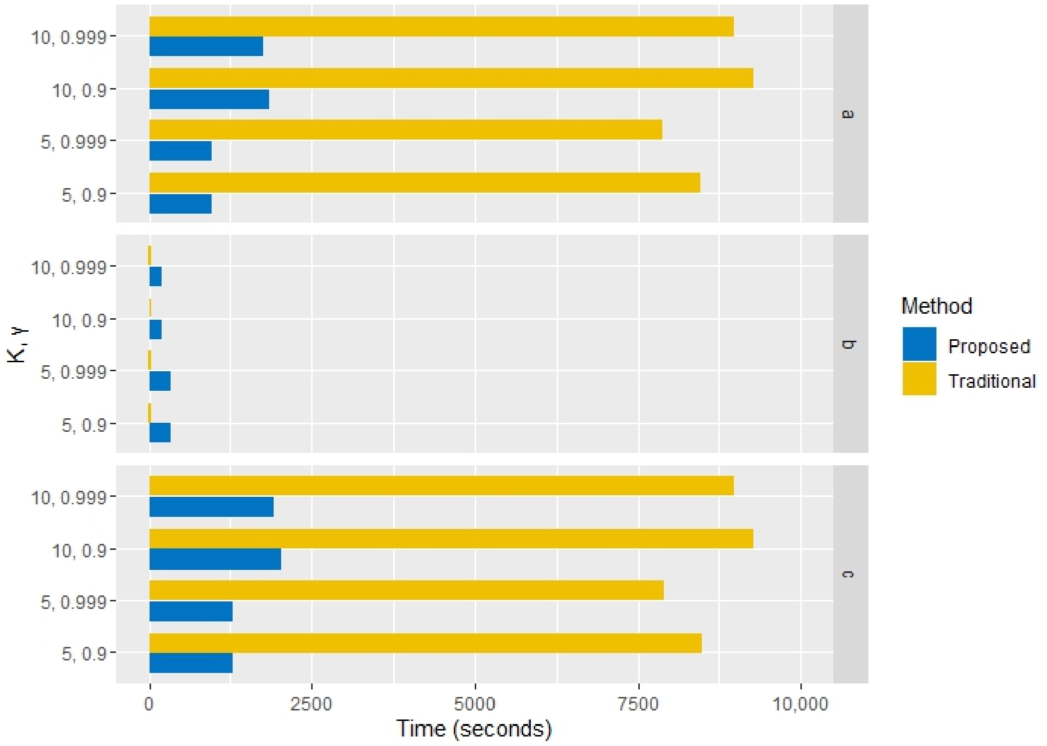

Figure 1. Time required to obtain (**a**) the parameter estimates and (**b**) the performance measures of the five estimators for both the traditional approach given in Section 2 and the approach proposed in Section 3. (**c**) Total time required to obtain the estimated model as well as the performance measures.

The computational time required to obtain the estimated model using the proposed approach was ±eight times faster than the traditional approach for the 5-fold cross-validation and ±five times faster for the 10-fold cross-validation. The time required to obtain the predicted responses and the performance measures on the testing dataset was less for the traditional approach. When using the traditional approach, the complete dataset was imported before the model estimation was performed. The dataset was never discarded, and hence it was readily available when prediction on the testing set commenced. This came at a cost since the memory used to store the complete dataset was unavailable at the model estimation stage. Although our proposed approach required the testing datasets to be imported again, the total time required to obtain the models, predict, and calculate the relevant performance measures was significantly less when using the approach proposed in Section 3.

Table 1. Difference in performance measures calculated for the approach proposed in Section 3 and the traditional approach given in Section 2 when $K = 5$ and $\mathbf{K = 10}$.

γ	**Estimator**	**MSE ($\hat{\beta}$)**	**MSE ($\hat{y}$)**	**SMAPE ($\hat{y}$)**
0.9	Ridge	3.866×10^{-15} $\mathbf{4.387 \times 10^{-11}}$	-2.776×10^{-17} $\mathbf{1.206 \times 10^{-13}}$	6.501×10^{-13} $\mathbf{3.270 \times 10^{-9}}$
0.9	Modified ridge	-1.784×10^{-9} $\mathbf{-9.465 \times 10^{-9}}$	3.679×10^{-10} $\mathbf{3.539 \times 10^{-10}}$	-4.567×10^{-7} $\mathbf{-6.069 \times 10^{-7}}$
0.9	Liu	2.568×10^{-8} $\mathbf{-5.866 \times 10^{-8}}$	1.825×10^{-9} $\mathbf{1.263 \times 10^{-9}}$	1.760×10^{-6} $\mathbf{-4.399 \times 10^{-6}}$
0.9	Modified Liu	3.077×10^{-9} $\mathbf{5.432 \times 10^{-9}}$	5.115×10^{-12} $\mathbf{-1.852 \times 10^{-12}}$	3.470×10^{-7} $\mathbf{2.999 \times 10^{-7}}$
0.9	Kibria–Lukman	-1.865×10^{-15} $\mathbf{-3.193 \times 10^{-16}}$	-1.388×10^{-17} $\mathbf{-6.939 \times 10^{-17}}$	1.421×10^{-13} $\mathbf{4.974 \times 10^{-14}}$
0.999	Ridge	4.184×10^{-12} $\mathbf{-4.048 \times 10^{-8}}$	2.637×10^{-16} $\mathbf{-2.637 \times 10^{-11}}$	2.540×10^{-12} $\mathbf{-1.835 \times 10^{-8}}$
0.999	Modified ridge	-6.462×10^{-6} $\mathbf{-4.155 \times 10^{-6}}$	-6.894×10^{-9} $\mathbf{5.182 \times 10^{-9}}$	-7.709×10^{-6} $\mathbf{-1.219 \times 10^{-7}}$
0.999	Liu	8.086×10^{-5} $\mathbf{-4.591 \times 10^{-6}}$	9.276×10^{-8} $\mathbf{8.133 \times 10^{-9}}$	-4.793×10^{-6} $\mathbf{-9.801 \times 10^{-6}}$
0.999	Modified Liu	3.448×10^{-6} $\mathbf{5.979 \times 10^{-6}}$	2.908×10^{-10} $\mathbf{1.749 \times 10^{-10}}$	1.503×10^{-6} $\mathbf{1.051 \times 10^{-6}}$
0.999	Kibria–Lukman	2.920×10^{-13} $\mathbf{4.710 \times 10^{-13}}$	2.776×10^{-17} $\mathbf{-1.388 \times 10^{-17}}$	1.386×10^{-13} $\mathbf{1.847 \times 10^{-13}}$

Furthermore, the difference between the performance measures calculated for the proposed approach and the traditional approach, where we performed the relevant matrix algebra from Section 2 using the entire dataset at once, was extremely close to zero. That is, the proposed approach was as accurate as the traditional approach where the mean squared error of the estimators, the mean prediction error, and the symmetric mean absolute percentage error of the predicted response on the testing dataset were concerned. The two experiments were also conducted on datasets where the level of multicollinearity was low, e.g., cases where $\gamma = 0.5, 0.6$. The results obtained were very similar to the results shown in Figure 1 and Table 1, and the conclusions arrived at also held for lower levels of multicollinearity.

4.2. Application

The dataset used in this application was obtained from the Bureau of Transportation Statistics, which provides information on U.S. transportation systems. The dataset contains information such as the origin and destination airport of various flights, the time at which flights are scheduled to arrive and depart, the delay in departing and arriving flights, the time and distance to destination airports, and the date and time of these flights, for major air carriers. We selected air traffic data from 2010 to 2020, resulting in a dataset with $\pm 6,000,000$ observations and ± 90 features. The response variable was the arrival delay in minutes. As in the simulation study, we considered two experiments. The first experiment aimed to compare the execution time of our proposed approach with the execution time of the traditional approach, where we performed the relevant matrix algebra from Section 2 using the entire dataset at once. The second experiment was to show that our algorithm was as accurate as the traditional approach. The performance measures used were the mean prediction error (MPE) and the symmetric mean absolute percentage error (SMAPE) of the predicted response on the testing dataset. The results obtained in the application are given in Figure 2 and Table 2.

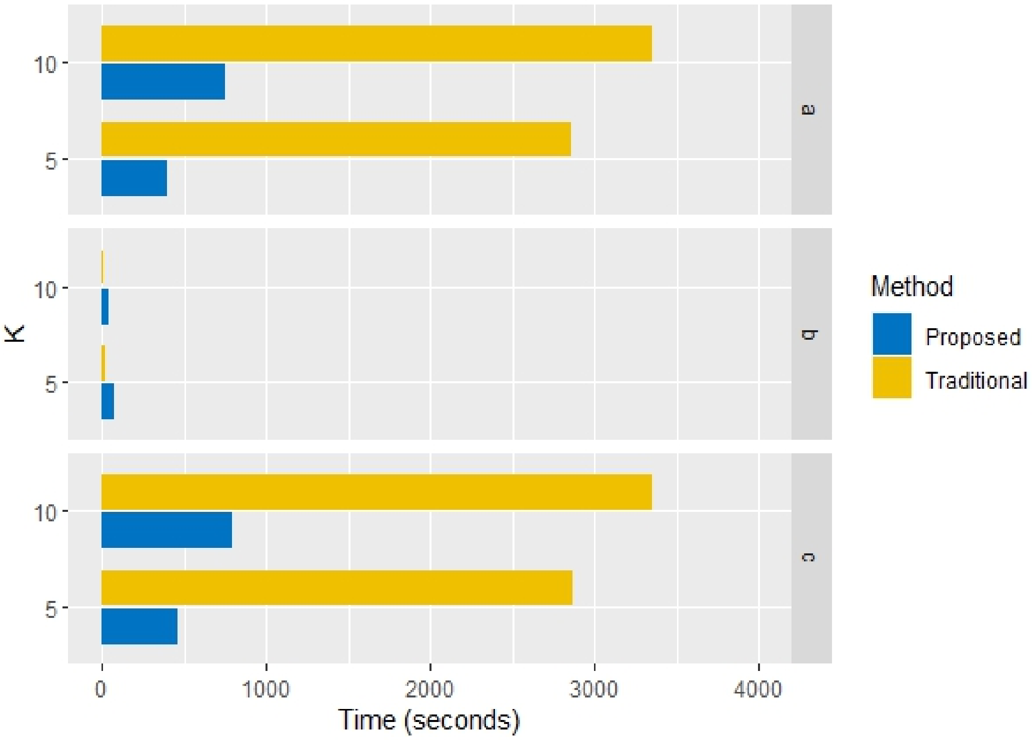

Figure 2. Time required to obtain (**a**) the parameter estimates and (**b**) the performance measures of the five estimators for both the traditional approach given in Section 2 and the approach proposed in Section 3. (**c**) Total time required to obtain the estimated model as well as the performance measures.

Table 2. Difference in performance measures calculated for the proposed approach and the traditional approach given in Section 2 when $K = 5$ and $\mathbf{K = 10}$.

Estimator	MPE ($\hat{y}$)	SMAPE ($\hat{y}$)
Ridge	4.025×10^{-5} $\mathbf{-2.948 \times 10^{-4}}$	6.719×10^{-7} $\mathbf{1.781 \times 10^{-5}}$
Modified ridge	1.803×10^{-3} $\mathbf{1.766 \times 10^{-3}}$	2.871×10^{-4} $\mathbf{1.673 \times 10^{-3}}$
Liu	5.253×10^{-4} $\mathbf{4.835 \times 10^{-4}}$	1.496×10^{-5} $\mathbf{3.623 \times 10^{-5}}$
Modified Liu	3.131×10^{-5} $\mathbf{9.429 \times 10^{-5}}$	9.764×10^{-7} $\mathbf{1.000 \times 10^{-5}}$
Kibria–Lukman	7.488×10^{-6} $\mathbf{7.397 \times 10^{-5}}$	5.760×10^{-7} $\mathbf{9.347 \times 10^{-6}}$

From the results given in Figure 2 and Table 2, it is clear that the total computational time required to obtain the models, predict, and calculate the relevant performance measures was significantly less when using the approach proposed in Section 3. Furthermore, the differences between the performance measures calculated for the proposed approach and the traditional approach were close to zero.

5. Conclusions and Future Work

Shrinkage methods are often used to mitigate the consequences of multicollinearity in linear regression models. These methods can easily be applied to small- or moderate-size datasets but encounter significant challenges, such as the memory and time required to obtain an optimal model in the big data domain. The goal of this paper was to propose a method for the efficient model estimation and validation of various well-known shrinkage estimators—mainly used to address multicollinearity—when considering large datasets. We proposed an algorithm that utilised an array of sufficient statistics for model estimation of various ridge- and Liu type estimators with closed-form solutions for the tuning parameter. Our algorithm also enabled model validation through K-fold cross-validation. A simulation study and an application on a real-world dataset illustrated the efficiency and accuracy of our approach. The estimators that we considered have a general mathematical form, and hence our approach is not limited to only the estimators presented in this paper. Furthermore, some ideas for future work include:

- Extending the methodology to generalised linear models.
- Extending the methodology to streaming data.
- Extending the methodology to robust estimators.
- Extending the methodology to estimators for feature selection, for which one may consider measures of validation and evaluation similar to those in [21].

Author Contributions: Conceptualization, S.d.P., M.A. and G.M.; funding acquisition, S.M.M. and M.A.; methodology, S.d.P., M.A. and G.M.; software, S.d.P.; supervision, M.A. and G.M.; writing—original draft, S.d.P.; writing—review and editing, S.d.P., M.A., G.M. and S.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research of M. Arashi is based upon research funded by the Iran National Science Foundation, grant no. 4015320.

Data Availability Statement: Data is available from the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, C.; Chen, M.H.; Schifano, E.; Wu, J.; Yan, J. Statistical methods and computing for big data. *Stat. Interface* **2016**, *9*, 399. [CrossRef] [PubMed]
2. Emerson, J.W.; Kane, M.J. Don't drown in the data. *Significance* **2012**, *9*, 38–39. [CrossRef]
3. Chan, J.Y.L.; Leow, S.M.H.; Bea, K.T.; Cheng, W.K.; Phoong, S.W.; Hong, Z.W.; Chen, Y.L. Mitigating the multicollinearity problem and its machine learning approach: A review. *Mathematics* **2022**, *10*, 1283. [CrossRef]
4. Shaheen, N.; Shah, I.; Almohaimeed, A.; Ali, S.; Alqifari, H.N. Some Modified Ridge Estimators for Handling the Multicollinearity Problem. *Mathematics* **2023**, *11*, 2522. [CrossRef]
5. Zhang, T.; Yang, B. An exact approach to ridge regression for big data. *Comput. Stat.* **2017**, *32*, 909–928. [CrossRef]
6. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
7. Lukman, A.F.; Ayinde, K.; Binuomote, S.; Clement, O.A. Modified ridge-type estimator to combat multicollinearity: Application to chemical data. *J. Chemom.* **2019**, *33*, e3125. [CrossRef]
8. Kejian, L. A new class of blased estimate in linear regression. *Commun. Stat.-Theory Methods* **1993**, *22*, 393–402. [CrossRef]
9. Lukman, A.F.; Kibria, B.; Ayinde, K.; Jegede, S.L. Modified one-parameter liu estimator for the linear regression model. *Model. Simul. Eng.* **2020**, *2020*, 9574304. [CrossRef]
10. Kibria, B.; Lukman, A.F. A new ridge-type estimator for the linear regression model: Simulations and applications. *Scientifica* **2020**, *2020*, 9758378. [CrossRef] [PubMed]
11. Kibria, B.G. Performance of some new ridge regression estimators. *Commun. Stat.-Simul. Comput.* **2003**, *32*, 419–435. [CrossRef]
12. Alkhamisi, M.; Khalaf, G.; Shukur, G. Some modifications for choosing ridge parameters. *Commun. Stat.-Theory Methods* **2006**, *35*, 2005–2020. [CrossRef]
13. Lukman, A.F.; Ayinde, K. Review and classifications of the ridge parameter estimation techniques. *Hacet. J. Math. Stat.* **2017**, *46*, 953–967. [CrossRef]
14. Muniz, G.; Kibria, B.G. On some ridge regression estimators: An empirical comparisons. *Commun. Stat.-Simul. Comput.* **2009**, *38*, 621–630. [CrossRef]

15. Arashi, M.; Saleh, A.M.E.; Kibria, B.G. *Theory of Ridge Regression Estimation with Applications*; John Wiley & Sons: New York, NY, USA, 2019.
16. Stein, C. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *Proc. 3rd Berkeley Symp. Math. Stat. Probab.* **1956**, *1*, 197–206.
17. Özkale, M.R.; Kaçiranlar, S. The restricted and unrestricted two-parameter estimators. *Commun. Stat.-Theory Methods* **2007**, *36*, 2707–2725. [CrossRef]
18. Hoerl, A.E.; Kannard, R.W.; Baldwin, K.F. Ridge regression: Some simulations. *Commun. Stat.-Theory Methods* **1975**, *4*, 105–123. [CrossRef]
19. Zou, H.; Li, R. One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Stat.* **2008**, *36*, 1509. [PubMed]
20. Saleh, A.M.E.; Arashi, M.; Saleh, R.A.; Norouzirad, M. *Rank-Based Methods for Shrinkage and Selection: With Application to Machine Learning*; John Wiley & Sons: New York, NY, USA, 2022.
21. Sechidis, K.; Azzimonti, L.; Pocock, A.; Corani, G.; Weatherall, J.; Brown, G. Efficient feature selection using shrinkage estimators. *Mach. Learn.* **2019**, *108*, 1261–1286. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.