A Novel Automatic Variable Ranking and Selection Algorithm for Severely Imbalanced Big Binary Data

by

Mehdi-Abderrahman Jabri

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Master of Science

in

Mathematics and Statistics

Guelph, Ontario, Canada

ABSTRACT

A NOVEL AUTOMATIC VARIABLE RANKING AND SELECTION ALGORITHM FOR SEVERELY IMBALANCED BIG BINARY DATA

Mehdi-Abderrahman Jabri
University of Guelph, 2021

Advisor:
Dr. Khurram Nadeem

This thesis develops a novel automatic variable ranking and selection algorithm for regularised ordinary logistic regression (OLR) models in the presence of severe class-imbalance and potentially involving large scale datasets. We also consider the possibility of strong correlation among a subset of signal and noise covariates. Our algorithm utilizes an ensemble of regularised OLR model fits, such as the Least Absolute Shrinkage and Selection Operator (LASSO), the two-stage Adaptive Lasso, and Ridge Regression, to obtain stable variable rankings. The algorithm also considers three automatic selection methods employed to recover a set of influential variables using derived rank scores from an ensemble of model fits. The simulation study results showed that our algorithm is robust against severe class-imbalance under the presence of highly correlated covariates, and consistently obtained stable variable rankings and each automatic selection method recovered high proportions of signal covariates whilst filtering out noise. We exemplify our methodology using a large volume of severely imbalanced high-dimensional wildland fire data, demonstrating the value of our methodology, which can also be used in other areas of application such as genomics and fraud detection.

Acknowledgments

I would like to express my sincerest appreciation to my supervisor Dr. Khurram Nadeem, a true professional, whom I have collaborated with since my undergraduate studies and has provided me the opportunity and funding to explore this topic. Dr. Khurram Nadeem has helped me tremendous ways, such as furthering my knowledge in the field of Statistics and improving the way I communicate research. I would also like extend my gratitude to Dr. Tony Desmond for serving on my examining committee and for also being a fantastic professor. There are several other professors I have had throughout my time here at the University of Guelph which I give many thanks towards and also to the other members of our faculty. Finally, words cannot express how much I thank my family for their dedication and support towards my education throughout my time in university.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis focuses on developing an automatic variable importance ranking and selection algorithm for ordinary logistic regression (OLR) models in the presence of severe class imbalance with large volumes of data and varying degrees of multicollinearity. Variable importance ranking and selection is a core topic in regression analysis, as it enhances the interpretation of the model of interest, while potentially improving predictive performance. The process of determining reliable covariate rankings becomes intractable when faced with severe class-imbalance with highly correlated covariates and involving a big dataset. Severe class-imbalance occurs when one class is only represented by a small number of examples (minority class) compared to the other (majority) class (García et al., 2012). When OLR models are fitted to a big dataset given a high class imbalance ratio such as 1:1000, there is a biased estimation towards the majority class leading to high misclassification rates (Ebenuwa et al., 2019).

This thesis further develops a novel automatic variable importance ranking and selection algorithm, introduced by Nadeem et al. (2020), who applied their method to flag influential variables in spatio-temporal wildland fire occurrence models. However, their approach is heuristic in nature, needs further development for automating the selection

process and was not investigated through a simulation study to thoroughly evaluate its performance. This thesis fills these gaps by i) rigorously describing the ranking algorithm; ii) developing automatic selection methods based on computed ranks; and iii) thoroughly assessing its performance by conducting a detailed simulation study. Variable importance in regression analysis can be ordinarily determined by the corresponding magnitude of the estimated scaled regression coefficients, then the covariates are ranked by importance according to these values (Murray and Conner, 2009). As for variable selection, this can be done automatically using conventional methods such as backward elimination, forward selection, and stepwise regression where hypothesis tests can be used to determine whether the relationship between a covariate and the response is significant (important). However, these methods under-estimate the standard errors of the coefficient estimates, the p-values tend to be low leading to overfitting and searches a large space of possible models (Kumar et al., 2019; Smith, 2018).

To circumvent the issues arising in stepwise selection methods, we opt for regularised OLR using the (logistic) lasso for our novel algorithm as a better alternative for automatic variable selection to improve predictive performance, and is implemented to prevent overfitting (Tibshirani, 1996; Kumar et al., 2019). The logistic-lasso modelling framework was also used in Nadeem et al. (2020) for variable selection and ranking. We also study our variable ranking and selection algorithm by implementing the two-stage adaptive lasso procedure (Zou, 2006) and logistic-ridge regression (Hoerl and Kennard, 1970). Although ridge regression does not perform automatic variable selection, we show that our algorithm is compatible with ridge regression to attain desirable variable ranking and selection results. Additionally, we investigate three different automatic selection methods as part of our algorithm to select a set of important covariates, since we cannot determine the set of important covariates solely based on the list of ranked covariates obtained from our algorithm. We note that Nadeem et al. (2020) did not investigate any automatic selection

methods and selected a set of important covariates by visual inspection instead.

The issue of severe class-imbalance can be relieved by using response-based sampling, which is done by sampling the same number of instances from both classes (Arezzo and Guagnano, 2018). Response-based sampling is commonly used in case-control studies in the field of epidemiology and "choice-based" studies in econometrics (Jiang et al., 2011). We specifically use response-based downsampling, where we retain all cases and randomly sample the number instances that correspond to cases from the controls to attain a balanced dataset. This process is repeated to generate many independent balanced datasets to create an ensemble of a large number of logistic-lasso model fits to attain a stable set of ranked covariates by their importance. Response-based downsampling is also implemented in other wildland fire occurrence studies such as Nadeem et al. (2020) and Woolford et al. (2011) and is discussed thoroughly in Hosmer et al. (2000). Implementing response-based sampling indirectly reduces the size of our dataset which allows us to efficiently carry out ensemble modelling and other computationally intensive statistical procedures (Fithian and Hastie, 2014).

Our simulation study design consists of four separate scenarios, two of which consider highly correlated covariates between noise and signal covariates, and the remaining contain non-correlated covariates. For each correlated and non-correlated scenario, we consider 100 and 200 covariates where 12 and 24 of the covariates correspond to non-zero regression coefficients respectively (signal covariates). We simulate data for three class-imbalance ratios each considering five sample sizes. We assess our novel algorithm using metrics such as true positive rate (TPR), false positive rate (FPR) and area under the ROC curve (AUC) corresponding to the variable selection performance of each regularised regression method (i.e standard logistic-lasso, adaptive lasso and ridge regression). These metrics are computed from an ensemble of a large number of model fits, and the variable selection performance for each automatic selection threshold technique is applied to the set of ranked co-

variates. We also investigate further into the models developed by Nadeem et al. (2020) as a case study, where we apply our algorithm to rank and select covariates corresponding to three spatio-temporal wildland fire occurrence models developed for the region of British Columbia. The three models developed are a human caused fire model (HCF), a predicted lightning caused fire (PLCF) model and observed lighting caused fire model (OLCF), given severely imbalanced high-dimensional big binary data over the course of 34 years, 1981-2014.

Overall, this thesis establishes that our automatic variable ranking and selection algorithm under the conditions of highly imbalanced big binary data is superior in recovering a high proportion of important covariates and screens out noisy covariates with the aid of implementing automatic selection methods. Our methodology also shows that we obtain more stable variable rankings by employing an ensemble of regularised regression models as opposed to the usual practice, where only a single model fit is employed to determine variable importance. We also make recommendations on which regularised regression and automatic selection methods should be implemented to attain optimal performance.

Chapter 2

Materials and Methods

We describe our variable ranking and selection algorithm in detail, in addition to three automatic selection methods studied in our algorithm. We also describe the outline of our simulation design as well as the data generation procedure.

2.1 Regularisation Techniques for OLR

This section introduces three different regularisation techniques for OLR considered within our simulation study.

2.1.1 Elastic Net Regularisation

Suppose we have n individuals, all with binary outcomes $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ and a p-dimensional vector of covariate values $\mathbf{x} = (x_1, \ldots, x_p)^\intercal$ such that $\underline{\mathbf{x}} = (\mathbf{1}, \mathbf{x})^\intercal$. The binary outcomes can be modelled using the OLR model with joint likelihood function $\mathcal{L} = \Pi_n \pi^y (1-\pi)^{1-y}$ and the OLR model is represented with a logit link function as: $\eta(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \boldsymbol{\beta}^\intercal \underline{\mathbf{x}}$, where $\pi = P(y_i = 1|\mathbf{x})$ is the probability of observing $y_i = 1$ for $i = 1, \ldots, n$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ is a vector of regression coefficients including the

intercept. The corresponding log-likelihood for OLR can be expressed as:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[(1 - y_i)\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i + \log(1 + e^{-\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i}) \right]. \tag{2.1}$$

We consider elastic net regularisation (Zou and Hastie, 2005) which penalises the OLR regression coefficients $\boldsymbol{\beta}$, using a combination of the l_1 and l_2 penalties, denoted as $\sum_{j=1}^{p} |\beta_j|$ and $\sum_{j=1}^{p} \beta_j^2$, respectively. We obtain the estimated elastic net coefficients $\widehat{\boldsymbol{\beta}}^{EN}$ by the following:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left[(1 - y_i)\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i + \log(1 + e^{-\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i}) \right] + \lambda \sum_{j=1}^{p} \left[\frac{1}{2}(1 - \alpha)\beta_j^2 + \alpha|\beta_j| \right] \right\} \tag{2.2}$$

where $\alpha \in (0, 1)$ controls the amount of weighting corresponding to l_1 and l_2 penalties. The penalization parameter λ controls the amount of shrinkage applied to the coefficients, and is selected via cross-validation using the `glmnet` package in `R` (Friedman et al., 2021).

2.1.2 The Least Absolute Shrinkage and Selection Operator (Lasso)

The lasso (Tibshirani, 1996) is a special case of the elastic net regularisation when $\alpha = 1$ in (2.2), and is a widely used method for variable selection, attempting to shrink irrelevant coefficients to zero in order to improve model interpretability and predictive skill. The lasso utilizes the l_1 penalty and its estimates $\widehat{\boldsymbol{\beta}}^{L}$ are given by the following:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left[(1 - y_i)\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i + \log(1 + e^{-\boldsymbol{\beta}^{\intercal}\underline{\mathbf{x}}_i}) \right] + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{2.3}$$

The lasso estimator does not enjoy the oracle properties under certain conditions, where an estimator performs as well as if the true underlying model were given in advance (Zou, 2006). If we denote $\widehat{\boldsymbol{\beta}}(\delta)$ as the coefficient estimator produced by fitting an oracle

procedure δ, then $\widehat{\boldsymbol{\beta}}(\delta)$ has the oracle properties as described by Zou (2006):

- Identifies the right subset model, $\{j : \widehat{\beta}_j \neq 0\} = \mathcal{A}$, where $\mathcal{A} = \{j : \beta_j^* \neq 0\}$
- Has the optimal estimation rate, $\sqrt{n}\left(\widehat{\boldsymbol{\beta}}(\delta)_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\right) \rightarrow_d N(\mathbf{0}, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\Sigma}^*$ is the covariance matrix if the true subset model is known.

Fan and Li (2001) state that the oracle properties do not hold for the lasso and the oracle properties should hold for a good procedure as argued by (Fan and Li, 2001; Fan et al., 2004).

2.1.3 Adaptive Lasso

The adaptive lasso was introduced by Zou (2006), which unlike the standard lasso, is consistent in variable selection, has the oracle properties and is able to eliminate bias in the estimated lasso coefficients. This method adds different weights to each coefficient in the l_1 penalty. The adaptive lasso coefficients can be estimated via the LARS algorithm, an iterative procedure proposed by Efron et al. (2004). This thesis opts for the two-stage procedure instead (see Zhou et al. (2009) for an example). The lasso estimates $\widehat{\boldsymbol{\beta}}^L$ are initially computed, then set $\gamma = 1$ (one can choose any $\gamma > 0$) and define a weight vector $\widehat{\mathbf{w}} = 1/|\widehat{\boldsymbol{\beta}}^L|^\gamma$, where the estimated adaptive lasso estimates $\widehat{\boldsymbol{\beta}}^{Ada}$ are given by the following:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left[(1 - y_i)\boldsymbol{\beta}^\intercal \underline{\mathbf{x}}_i + \log(1 + e^{-\boldsymbol{\beta}^\intercal \underline{\mathbf{x}}_i}) \right] - \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}. \tag{2.4}$$

With regards to the oracle properties, Zou (2006) shows that these properties hold under mild conditions for generalized linear models (GLMs). We first consider the penalized log-likelihood with the weighed l_1 penalty, where the likelihood belongs to the exponential family with canonical parameter θ (Zou, 2006). Then the generic family form is

represented by McCullagh and Nelder (1989) as:

$$f(y|x,\theta) = h(y)e^{(y\theta-\phi(\theta))}, \tag{2.5}$$

such that $\theta = \boldsymbol{x}^\intercal\boldsymbol{\beta}$. Suppose that $\widehat{\boldsymbol{\beta}}(mle)$ is the maximum likelihood estimates in the GLM and define a weight vector $\widehat{\boldsymbol{w}} = 1/|\widehat{\boldsymbol{\beta}}(mle)|^\gamma$ for some $\gamma > 0$ (Zou, 2006). Then the adaptive lasso estimates $\widehat{\boldsymbol{\beta}}^{*(n)}(glm)$ are given by minimizing:

$$\sum_{i=1}^{n}(-y_i(x_i^\intercal\boldsymbol{\beta})) + \phi(x_i^\intercal\boldsymbol{\beta}) + \lambda_n\sum_{j=1}^{p}\widehat{w}_j|\beta_j|. \tag{2.6}$$

If an appropriate value of λ_n is selected then $\widehat{\boldsymbol{\beta}}^{*(n)}(glm)$ enjoys the oracle properties. If we let $A_n^* = \{\widehat{\boldsymbol{\beta}}_j^{*(n)}(glm) \neq 0\}$ and suppose that $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$; then under some mild regularity conditions $\widehat{\beta}^{*(n)}(glm)$ must satisfy the following:

1. Consistency in variable selection: $\lim_n P(A_n^* = A) = 1$, where $A = \{j : \beta_j^{*(n)}(glm) \neq 0\}$
2. Asymptotic normality: $\sqrt{n}(\widehat{\boldsymbol{\beta}}_A^{*(n)}(glm) - \boldsymbol{\beta}_A^*) \to_d N(\mathbf{0}, \boldsymbol{I}_{11}^{-1})$, where $\boldsymbol{I}_{11}^{-1}$ is the Fisher information and is a $p_0 \times p_0$ matrix, with the true submodel known.

Zou (2006) shows that iterative procedures such as the Newton-Raphson method can be used to solve for $\widehat{\boldsymbol{\beta}}_A^{*(n)}(glm)$. We also note that in the ordinary least squares (OLS) case, $\widehat{\boldsymbol{\beta}}$ is not required to be root-n consistent for the adaptive lasso for the oracle properties to hold (Zou, 2006).

2.1.4 Ridge Regression

Ridge regression utilizes the l_2 penalty technique that seeks to reduce bias due to overfitting and improve model prediction, but unlike the lasso this does not perform variable selection, rather the regression coefficients are shrunk towards zero. Ridge regression

is also a special case of the elastic-net when $\alpha = 0$. Ridge regression was originally introduced to resolve the issue of instability of OLS estimates due to multicollinearity (Wu, 2020). The ridge regression estimates $\widehat{\boldsymbol{\beta}}^R$ are given by following:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left[(1 - y_i) \boldsymbol{\beta}^\intercal \underline{\mathbf{x}}_i + \log(1 + e^{-\boldsymbol{\beta}^\intercal \underline{\mathbf{x}}_i}) \right] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2.7}$$

Since ridge regression does not perform variable selection, oracle properties do not hold. On the other hand, Broken adaptive ridge regression (BAR) (Zhao et al., 2018), which performs variable selection and its corresponding estimates possess the oracle properties.

2.2 Class Imbalance and Response-Based Sampling

2.2.1 Class-Imbalance

Severe class-imbalance must be resolved prior to fitting OLR models, since leaving this problem unaddressed leads to bias towards the majority class during prediction. It is possible the minority class examples are ignored by the classifier since it becomes more difficult to distinguish between minority class examples (Fernández et al., 2013). There are several practical scenarios with severe class imbalance where the minority class is the main target for the researchers, for instance credit card fraud detection or wildland fire occurrences. In this context, OLR models would not accurately predict whether an example is fraud or whether a wildland fire occurred due to estimation bias towards the majority class and the model fitting process would be computationally expensive. There are three main approaches taken to tackle class-imbalance for both standard-learning algorithms and ensemble techniques described by Fernández et al. (2013):

1. Data level solutions: Rebalance the class distribution by sampling the data space to remove the effect of class-imbalance.

2. Algorithmic level solutions: Adapt specific learning algorithms to reinforce the learning towards the positive class.

3. Cost sensitive solutions: These incorporate methods at the data level, algorithmic level, or both levels jointly, which consider higher miss-classification cost for positive class examples with respect to examples of negative class, trying to minimize higher cost errors.

This thesis uses response-based sampling, a data level solution as it is much simpler and efficient to implement because we can pre-process the data before model fitting on different classifiers for example, and this method is also independent of which classifier we use (Xie and Manski, 1989; Fernández et al., 2013). Repeated sampling procedures can be implemented for the purpose of relieving class-imbalance, also substantially reducing the amount of examples in a dataset, which is useful especially when performing already computationally intensive statistical procedures. The limitations that arise in some algorithmic/cost-sensitive solutions include; i) modifying learning algorithms (classifier) that may pose to be difficult, also extra parameters may be needed to tune, and ii) the computational expense that arises with big data, especially for matrix computations (García-Pedrajas et al., 2012; Yu et al., 2018).

2.2.2 Response-Based Sampling

Response-based sampling is commonly used in case-control studies and we specifically use it for relieving severe-class imbalance, where we *downsample* controls ($y = 0$). Initially, all n_1 cases ($y = 1$) are retained and n_1 controls are randomly sampled to create a *balanced* sample. Xie and Manski (1989) define the likelihood of an observation via response-based sampling as: $P(\mathbf{x}|y)\frac{Ny}{N}$, where $\frac{Ny}{N}$ is the sampling proportion for response y. Response-based sampling changes the OLR modelling framework by inducing an offset

term added to the regression intercept as shown in section 6.3 of Hosmer et al. (2000). Let s denote the selection ($s = 1$) or non-selection ($s = 0$) of a subject. The likelihood with n_1 cases and n_0 controls is:

$$\prod_{i=1}^{n_1} P(\underline{\mathbf{x}}_i|y_i = 1, s_i = 1) \prod_{i=1}^{n_0} P(\underline{\mathbf{x}}_i|y_i = 0, s_i = 0). \tag{2.8}$$

Using Bayes' theorem, we can rewrite (2.8) for an individual term in the likelihood function as:

$$P(\underline{\mathbf{x}}|y, s = 1) = \frac{P(y|\underline{\mathbf{x}}, s = 1)P(\underline{\mathbf{x}}|s = 1)}{P(y|s = 1)}. \tag{2.9}$$

When $y = 1$, we represent $P(y|\mathbf{x}, s = 1)$ as:

$$P(y = 1|\underline{\mathbf{x}}, s = 1) = \frac{P(y = 1|\underline{\mathbf{x}})P(s = 1|\underline{\mathbf{x}}, y = 1)}{P(y = 0|\underline{\mathbf{x}})P(s = 1|\underline{\mathbf{x}}, y = 0) + P(y = 1|\underline{\mathbf{x}})P(s = 1|\underline{\mathbf{x}}, y = 1)} \tag{2.10}$$

If we assume that the selection of cases and controls is independent of the covariates with respective probabilities;

$$\tau_1 = P(s = 1|y = 1, \underline{\mathbf{x}}) = P(s = 1|y = 1)$$

and

$$\tau_0 = P(s = 1|y = 0, \underline{\mathbf{x}}) = P(s = 1|y = 0).$$

We can represent the OLR model $\eta(\pi(\underline{\mathbf{x}}))$ by substituting τ_0 and τ_1 for $P(y = 1|\underline{\mathbf{x}})$ into (2.10) yielding:

$$P(y = 1|\underline{\mathbf{x}}, s = 1) = \frac{\tau_1\pi(\underline{\mathbf{x}})}{\tau_0\left[1 - \pi(\underline{\mathbf{x}})\right] + \tau_1\pi(\underline{\mathbf{x}})}. \tag{2.11}$$

Dividing the numerator and denominator of the right hand side of (2.11) by $\tau_0\left[1 - \pi(\underline{\mathbf{x}})\right]$, results in a OLR model with an intercept term $\beta_0^* = \log(\tau_1/\tau_0) + \beta_0$, with $\tau_1 = 1$ and τ_0

is the ratio of the number of controls to the number of cases in the entire dataset (Nadeem et al., 2020). By re-balancing the class distribution through response-based sampling the resultant logistic model is represented by: $\eta(\pi^*(\underline{\mathbf{x}})) = \log(\tau_1/\tau_0) + \beta^\intercal \underline{\mathbf{x}}$, where $\pi^*(\underline{\mathbf{x}}) = P(y = 1|\underline{\mathbf{x}}, s = 1)$. The likelihood function in (2.8) becomes:

$$\mathcal{L}^*(\boldsymbol{\beta}) \prod_{i=1}^{n} \left[\frac{P(\underline{\mathbf{x}}_i)}{P(y_i|s_i = 1)} \right] \tag{2.12}$$

where $\mathcal{L}^*(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi^*(\underline{\mathbf{x}}_i)^{y_i}[1 - \pi^*(\underline{\mathbf{x}}_i)]^{1-y_i}$ (Hosmer et al., 2000). Although response-based sampling tackles severe class imbalance by re-balancing the class distribution, it can also substantially reduce the volume of data, allowing for repeated use in order to fit many OLR models for the purpose variable importance ranking which would not be feasible with a large scale of data.

2.3 Automatic Variable Ranking and Selection Methods

In this section we describe our novel variable ranking and selection algorithm in detail and introduce automatic thresholding methods for automatically selecting important covariates from variable rankings. A core feature of our algorithm is to create an ensemble of regularised OLR models in order to generate a set of covariates ranked by their relative importance, aiding the process of selecting relevant and filtering noise covariates. Each model across the ensemble is fitted to a separate (balanced) dataset achieved by response-based sampling by exploring the sample space $P(\mathbf{x}|y) = 0$, allowing for variation (diversity) of covariate selection between each model. We would be able to observe that covariates of true importance will be frequently selected unlike the noise covariates. The supe-

riority of creating an ensemble of regularised OLR models is that when performing automatic variable selection (for example the logistic-lasso) there is a high possibility that a single model fit performs worse (by selecting more noise or filtering important covariates) than the ensemble.

Our algorithm is summarised in the four main steps:

1. The implementation of re-sampling methods (response-based sampling) to create an ensemble of $M = 500$ balanced datasets.

2. The use of regularised OLR to perform automatic variable selection fitted on each balanced dataset.

3. Computing the average of the standardized regression coefficients across the M model fits. This is done to get a more stable and reliable importance measure for each covariate, especially with a large value of M (Zhang et al., 2017; Nadeem et al., 2020).

4. Applying an automatic selection technique using the resultant rank metrics to obtain a subset of ranked covariates by their relative importance.

There also includes a intermediate stage in the algorithm that aids the process of selecting a reduced set of important covariates. First we let $\widehat{\boldsymbol{\beta}}^{\omega} = (\widehat{\beta}_1^{\omega}, \ldots, \widehat{\beta}_j^{\omega})$ represent the vector of estimated regularised OLR coefficients corresponding to either the lasso, adaptive lasso and ridge regression models. We compute the index $P_{Drop_i} = \sum_{j=1}^{M} I_{\widehat{\beta}_{i,j}^{\omega}=0}/M$, which is the proportion of times covariate x_i is dropped from M model fits (Nadeem et al., 2020).

2.3.1 Automatic Variable Ranking Algorithm

The variable ranking and selection algorithm is described in detail below, from steps 1-6:

1. Compute standardized coefficients $\widehat{\beta}_{i,j}^{\omega\Delta} = s.d(x_i)\widehat{\beta}_{i,j}^{\omega}$ for regression covariates $\mathbf{x} = (x_1, x_2, \ldots, x_P)^\intercal$, where $s.d(x_i)$ is the standard deviation of the i^{th} covariate and the standardized coefficients correspond to the transformed covariates $z_k = \frac{x_k - \bar{x}_k}{s.d(x_k)}$

2. We rank the absolute valued standardized coefficients $\widehat{\beta}_{i,j}^{\omega\Delta} = \left(|\widehat{\beta}_{1,j}^{\omega\Delta}|, \ldots, |\widehat{\beta}_{P,j}^{\omega\Delta}|\right)$ letting $(R_{1,j}, \ldots, R_{P,j})$, be their respective ranks where $1 \leqslant R_{i,j} \leqslant P$, and the covariates with the highest and lowest values are ranked at P and 1 respectively. Sorting the $\widehat{\boldsymbol{\beta}}_j^{\omega\Delta}$ vector results in $\widetilde{\boldsymbol{\beta}}_j^{\omega\Delta} = \left(|\widehat{\beta}_{i^{(1)},j}^{\omega\Delta}|, \ldots, |\widehat{\beta}_{i^{(P)},j}^{\omega\Delta}|\right)$ such that $|\widehat{\beta}_{i^{(h)},j}^{\omega\Delta}| \leqslant |\widehat{\beta}_{i^{(h+1)}}^{\omega\Delta}|$ and $x_{i^{(h)}}$ is the covariate with assigned rank h. For example, if x_5 has an assigned rank value of 10 then $i^{(10)} = 5$. For a given rank position h, there can be differences of $x_{i^{(h)}}$ between model fits for $j = 1, \ldots, M$.

3. The rank score for a covariate amongst M model fits is given by:

$$Rank(x_i) = \frac{1}{M}\sum_{h=1}^{P}\sum_{j=1}^{M} hI_{R_{i,j}=h}, \tag{2.13}$$

where I_A is an indicator function corresponding to event A and h is the rank position.

Theorem 1:

i) $1 \leqslant Rank(x_i) \leqslant P$

ii) $\sum_{i=1}^{P} Rank(x_i) = \sum_{h=1}^{P} h = \frac{P(P+1)}{2}$

Proof: i) We rewrite $Rank(x_i)$ as:

$$Rank(x_i) = \frac{1}{M}\sum_{j=1}^{M}\sum_{h=1}^{P} hI_{R_{i,j}=h}.$$

Then we must have $1 \leqslant \sum_{i=1}^{P} hI_{R_{i,h}=h} \leqslant P$ for a given j. The average over

the index j must also satisfy the same constraints.

ii) Now consider the sum:

$$\begin{aligned}
\sum_{i=1}^{P} Rank(x_i) &= \frac{1}{M}\sum_{i=1}^{P}\sum_{h=1}^{M}\sum_{h=1}^{P} hI_{R_{i,j}=h} \\
&= \frac{1}{M}\sum_{j=1}^{M}\sum_{h=1}^{P} h\left(\sum_{i=1}^{P} I_{R_{i,j}=h}\right), \\
&= \frac{1}{M}\sum_{j=1}^{M}\sum_{h=1}^{P} h(1), \\
&= \frac{1}{M}\sum_{j=1}^{M}\left(\sum_{h=1}^{P} h\right), \\
\sum_{i=1}^{P} Rank(x_i) &= \sum_{h=1}^{P} h = \frac{P(P+1)}{2}.
\end{aligned}$$

4. Many covariates are usually shrunk to zero for a logistic-lasso fit with a large number of covariates due to the l_1 regularisation penalty. The covariates with coefficients shrunk to zero from a logistic-lasso fit are not to be included in the rankings as they have no effect on the response. First we define the set of selected covariates as:

   Definition 1 (Selected Covariates):

$$\boldsymbol{\chi} = \{x_i \in \mathbf{x} : \sum_{j=1}^{M} |\widehat{\beta}_{i,j}^{\Delta\omega}| > 0\}.$$

   We denote the cardinality of this set by $Q(\leqslant P)$. For any $1 \leqslant Q \leqslant P$ there is a possibility for a covariate (say x^*) to have a zero coefficient in some j^{th} lasso/adaptive-lasso fit. For example, if x^* is ranked at position 10 in a particular fit, then it's rank contribution in (2.13) from this fit is $h = 10$, even though it's estimated coefficient

is 0, however x^* should be penalized as it was dropped from selection in the j^{th} fit, therefore we rewrite $Rank(x_i)$ as:

$$Rank(x_i) = \frac{1}{M}\sum_{j=1}^{M}\sum_{h=1}^{P} hI_{R_{i,j}=h}I_{|\hat{\beta}_{i,j}^{\omega\Delta}|>0}, \tag{2.14}$$

where $Rank(x_i) \leqslant P$, and $\sum_{i=1}^{P} Rank(x_i) \leqslant \sum_{h=1}^{P} h$.

5. We compute modified ranks for each covariate due to the behaviour of the logistic-lasso/adaptive lasso given a large set of covariates in the OLR model, where several estimated coefficients are sent to zero and should not be included in the final ranks. Covariates with estimated coefficients that are zeroed-out for almost all M model fits would cause all regression coefficients ranked at a certain position to become zero. This leads us to present the following definition and an associated theorem:

Definition 2 (Effective Maximum Rank):

$$P_{EF} = max\{h \in (1,\ldots,P) : \sum_{j=1}^{M}\sum_{i=1}^{P} |\hat{\beta}_{i,j}^{\omega\Delta}|I_{R_{i,j}=h} > 0\},$$

where $\sum_{i=1}^{P} |\hat{\beta}_{i,j}^{\omega\Delta}|I_{R_{i,j}=h} = |\hat{\beta}_{i^{(h)},j}^{\omega\Delta}|$ for a given value h, and $P_{EF} \leqslant P$.

Theorem 2:

Let $S_h = \sum_{j=1}^{M} |\hat{\beta}_{i^{(h)},j}|$ and assume that $S_{\tilde{h}} = 0$ for some $\tilde{h} < P$, then we have $S_h = 0$ for all $h \geqslant \tilde{h}$.

Proof:

$S_{\tilde{h}} = 0$ implies that $|\hat{\beta}_{i^{(\tilde{h})},j}^{\omega\Delta}| = 0$ for all j. This also implies that $|\hat{\beta}_{i^{(\tilde{h}+1)},j}^{\omega\Delta}|$ must all be zero because, for any given j, we have by definition: $|\hat{\beta}_{i^{(h)},j}^{\omega\Delta}| \leqslant |\hat{\beta}_{i^{(h+1)},j}^{\omega\Delta}|$. Therefore, we must have $S_{\tilde{h}+1} = 0$; and $S_{\tilde{h}+k} = 0$ for all $1 \leqslant k \leqslant$

$(P - \tilde{h})$.

The computation of P_{EF} ignores all x_i that attain a $P_{Drop} = 1$. We now define the following:

Definition 3: Let $\boldsymbol{B} = [|\widetilde{\boldsymbol{\beta}}_1^{\Delta\omega}|, |\widetilde{\boldsymbol{\beta}}_2^{\Delta\omega}| \ldots, |\widetilde{\boldsymbol{\beta}}_M^{\Delta\omega}|]^\intercal$ bet a $M \times P$ matrix whose entries are (j, h) are given as $|\beta^{\Delta\omega}_{i^{(h)},j}|$.

Considering Theorem 2, the modified version of $Rank(x_i)$ for a maximum effective rank P_{EF} is given as:

$$Rank(x_i) = \frac{1}{M} \sum_{j=1}^{M} \sum_{h=1}^{P_{EF}} h I_{i=i^{(h)}} I_{\left|\hat{\beta}^{\omega\Delta}_{i^{(h)},j}\right| > 0}, \tag{2.15}$$

where $x_i \in \boldsymbol{\chi}$ and the second summation on the right hand side of (2.15) is taken over the first P_{EF} columns of $\boldsymbol{B}$. We see that (2.15) possesses the properties:

(a) $0 < Rank(x_i) \leqslant P_{EF}$

(b) $\sum_{i=1}^{P} Rank(x_i) \leqslant \sum_{h=1}^{P_{EF}} h$ where $Rank(x_i) = 0$ for $x_i \in \boldsymbol{\chi}^c$.

6. Utilize an automatic selection technique to determine a threshold corresponding to $Rank(x_i)$ and/or P_{Drop} values to retain important covariates. Since ridge regression does not perform automatic variable selection, P_{Drop} is not computed and we then have $P_{EF} = P$. The automatic selection techniques must then retain the important covariates based on $Rank(x_i)$ in (2.15).

2.3.2 Automatic Selection Techniques

Acquiring a list of covariates ranked by their relative importance obtained from the algorithm described in Section 2.3.1 only completes the task of "variable ranking". It must be known which and how many of these covariates are influential predictors in our OLR

models. By simply examining the rank metrics ($Rank(x_i)/P_{Drop}$ values) in order to determine a threshold that differentiates between the set of influential and irrelevant covariates is impractical, as it is inefficient and ambiguities may arise in the selection process. We address these potential issues by utilizing automatic threshold based selection methods within our algorithm to obtain important covariates. We study three different threshold methods which are to be applied on $Rank(x_i)$ and P_{Drop} values, where one of methods utilizes clustering to find a set of important covariates based on the relationship between the $Rank(x_i)$ and P_{Drop} values.

Maximum Difference Threshold Method

This method finds the largest rank score difference between two consecutive covariates in an ordered list of rankings, and selects the covariates that rank above the position where the largest difference occurs (say d). The maximum difference method applied on $Rank(x_i)$ is shown as:

1. Find the rank position d, where the largest difference between the two consecutive aggregated rank scores occur, where:

$$d = max\{Rank(x_{i^{(h)}}) - Rank(x_{i^{(h+1)}}) : h = 1, \ldots, H-1\},$$

   where $Rank(x_{i^{(h)}}) - Rank(x_{i^{(h+1)}}) \geqslant 0$ for all $h = 1, \ldots, H-1$, and H is the total number of rank positions.

2. Then select all covariates at above and including rank position d: $x_{i^{(1)}}, \ldots, x_{i^{(d)}}$

Likewise we apply this method to the P_{Drop} values corresponding to each covariate x_i such that $P_{Drop_i} \neq 1$ for $i = 1, \ldots, p$.

$$d = max\{|P_{Drop}(x_{i^{(h)}}) - P_{Drop}(x_{i^{(h+1)}})| : h = 1, \ldots, H-1\},$$

where $|P_{Drop}(x_{i^{(h)}}) - P_{Drop}(x_{i^{(h+1)}})| \geqslant 0$ for all $h = 1, \ldots, H-1$. We consider the absolute valued difference of P_{Drop} values because it is possible the covariate may have a lower P_{Drop} value than the subsequent ranked covariate.

Change Point Detection via Hypothesis Testing

Alternatively, we can also find a single change point of $Rank(x_i)$ or P_{Drop} values using hypothesis testing. Suppose we have an ordered sequence of values $u_{1:v} = \{u_1, \ldots, u_v\}$ and if there exists an index $\tau \in \{1, \ldots, v-1\}$ such that some statistical property of $\{u_1, \ldots, u_\tau\}$ and $\{u_{\tau+1}, \ldots, u_v\}$ differ, then a change point has occurred (Killick and Eckley, 2014). We then denote the null hypothesis as H_0 which corresponds to no change point detected ($m = 0$) and the alternative hypothesis H_1 where a single change point has been detected ($m = 1$) (Killick and Eckley, 2014). The hypothesis is tested using the general likelihood ratio based approach, used in Hinkley (1970), to test for a change point in the mean of normally distributed observations (Killick and Eckley, 2014). The maximum likelihood estimates are computed under both the null and alternative hypothesis for the likelihood ratio method. Under the null hypothesis the maximum likelihood is $\log p(u_{1:v}|\widehat{\theta})$, where $p(\cdot)$ is the probability density function associated with the distribution of the $u_{1:v}$ and $\widehat{\theta}$ is the maximum likelihood estimate of the parameters. Then the maximum log-likelihood estimate under the alternative hypothesis where we consider the change point $\tau_1 \in \{1, \ldots, v-1\}$ is shown by Killick and Eckley (2014) as:

$$ML(\tau_1) = \max_{\tau_1} \ \log p(u_{1:\tau_1}|\widehat{\theta}_0) + \log p(u_{(\tau_1+1):v}|\widehat{\theta}_a), \tag{2.16}$$

the test statistic is then:

$$\Lambda = 2\left[ML(\tau_1) - \log p(u_{1:v}|\widehat{\theta})\right]. \tag{2.17}$$

We reject the null hypothesis (detect a change point) if $\Lambda > c$, where c is a selected threshold, and $\widehat{\tau}_1$ is the estimate of τ_1 that maximizes $ML(\tau_1)$. Suggestions for selecting the value c are discussed further in detail by Guyon and Yao (1999); Lavielle (2005); Birgé and Massart (2007). We utilize the `mean.cpt()` function via the `changepoint` package (Killick et al., 2016) in `R`, to find a single change point in the mean $Rank(x_i)$ or P_{Drop} values.

Agglomerative Hierarchical Clustering

The last automatic threshold selection method we study is via agglomerative hierarchical clustering with complete-linkage. We apply this method to automatically recover a set of important covariates by examining the relationship between a covariate's $Rank(x_i)$ score and it's corresponding P_{Drop} value. The algorithm as described by Johnson et al. (2002) proceeds as follows:

1. Suppose we have N clusters with only one unit and represented by the distance matrix $\boldsymbol{D}_{N\times N} = \{d_{ik}\}$ for $i, k = 1, \ldots, K$, where $K = N$ and K is the number of desired clusters, here we select $K = 3$.

2. Compute and identify the largest distance between two clusters U and V via max-linkage: $d_{(UV)W} = max\{d_{(UW)}, d_{(VW)}\}$.

3. Merge U and V into a new cluster (U, V) and update $\boldsymbol{D}_{N\times N}$ by deleting rows/columns corresponding to each cluster, and inserting a row/column for (U, V) with distance to all other clusters.

4. Repeat steps 2. and 3. for $(N-1)$ times until all units are in a single cluster.

We denote cluster C_1 to the covariates of high importance, C_2 as medium important and C_3 as covariates of low importance. We select clusters the final set of covariates based on the condition; if the number of covariates contained in C_2 is less than $1/2$ the amount of covariates contained in C_1 then we select covariates in C_1 and C_2, otherwise we select C_1. Hierarchical clustering with complete-linkage is implemented in the `R` package `cluster` (Maechler et al., 2019).

We incorporate these automatic threshold methods within our algorithm along with the different variations of regularised OLR (lasso, adaptive lasso and ridge regression) and examine these methods via a simulation study. This is conducted to make conclusions regarding which regularised OLR and automatic threshold method together attain optimal variable ranking and selection performances given a large volume of data with severe class-imbalance and highly correlated covariates.

Chapter 3

Simulation Study Design

We designed four separate simulation models to execute our variable ranking and selection algorithm. For two of the simulation models, we consider data with correlated signal covariates where a subset of them are also correlated with the noise covariates, while the remaining two simulation models do not possess correlated covariates. We denote p as the number of total covariates in our OLR models where r of them are signal. The simulation models are defined in Table 3.1.

Table 3.1: Description of each simulation model

Simulation Model	p	r	Correlated Covariates
MUNCOR-12	100	12	no
MUNCOR-24	200	24	no
MCOR-12	100	12	yes
MCOR-24	200	24	yes

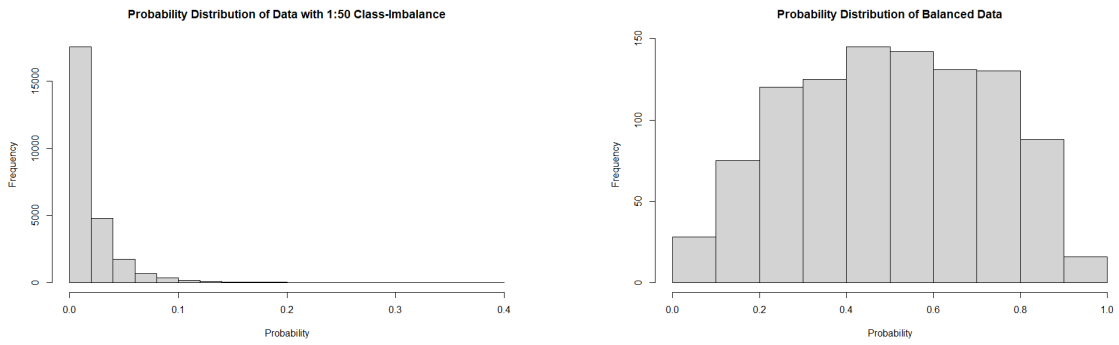
For MCOR-12 and MCOR-24 we generate correlated binary and nominal responses via the `SimCorMultRes` package (Touloumis, 2019) in `R`. We describe the methodology underlying in this package in Section 3.2. We create balanced datasets of size n_b using response-based sampling performed on original datasets (say, size of n^*) generated un-

der each simulation model (MUNCOR-12, MCOR-12, MUNCOR-24 and MCOR-24) with varying class-imbalance ratios I_R.

Table 3.2: Initial size of original dataset n^* generated under each class-imbalance ratio I_R needed to achieve each target balanced sample size n_b.

n_b	I_R		
	1:50	1:100	1:1000
1000	25,500	50,500	500,500
2000	51,000	101,000	1,001,000
3000	76,500	151,500	1,501,500
4000	102,000	202,000	2,002,000
5000	127,500	252,500	2,502,500

For each regularised OLR model, we consider two values of the penalization parameter λ, namely "lambda.min" which is the value of λ that gives the minimum mean cross-validation error based off the performance measure AUC. We also consider "lambda.1se", the value of λ where the cross-validation error is 1 standard deviation away from lambda.min, resulting in a more parsimonious model with reasonable predictive skill (Friedman et al., 2010). The effect of class-imbalance skews the distribution of class probabilities towards the majority class effecting the predictive skills of our models. Figure 3.1 below depicts the distribution of simulated class probabilities for the MUNCOR-12 model with an initial class-imbalance of 1:50 and after adjusting the intercept with the induced offset on the balanced dataset described in Section 2.2.



(a) (b)

Figure 3.1: Distribution of class probabilities with imbalanced data (a) and with balanced data after adjusting OLR intercept with offset (b).

3.1 Data Generation of Independent Covariates

In the simulation models MUNCOR-12 and MCOR-12, the non-zero OLR coefficients $\beta_1, \ldots, \beta_{12}$ respectively correspond to $\beta^{\{12\}} = (-0.9, -0.8, -0.7, -0.6, -0.5, -0.4, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5)$. Under MCOR-24 and MUNCOR-24, the non-zero OLR coefficients $\beta_1, \ldots, \beta_{24}$ respectively correspond to $\beta^{\{24\}} = (-1.20, -1.10, -1.00, -0.90, -0.80, -0.70, -0.60, -0.50, -0.40, -0.30, -0.20, -0.10, 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.05, 1.15)$.

The data generation process of the covariates corresponding to MUNCOR-12 are described below:

Signal Covariates

$X_1, \ldots, X_6 \sim bin(0.5)$ and $X_7, \ldots, X_{12} \sim N(0, 1)$.

Noise Covariates

$X_{13}, \ldots, X_{18} \sim bin(0.5)$ and $X_{19}, \ldots, X_{100} \sim unif(0, 1)$.

The covariates $X_7, \ldots, X_{12}$ are mapped to the [0,1] domain via a logit transformation. This is done to measure all relationships between covariates and the response on the same scale,

since a covariate may have a less or more impact to the response compared to the rest, due to the scale it is measured on, and will effect the interpretation of relative variable importance. The data generation process of the covariates corresponding to MUNCOR-24 are described below:

Signal Covariates

$$X_1, \ldots, X_6 \text{ and } X_{13}, \ldots, X_{18} \sim bin(0.5);\ X_7, \ldots, X_{12} \text{ and } X_{19}, \ldots, X_{24} \sim N(0,1).$$

Noise Covariates

$$X_{25}, \ldots, X_{200} \sim unif(0,1).$$

The covariates $X_7, \ldots, X_{12}$ and $X_{19}, \ldots, X_{24}$ are mapped to the [0,1] domain via a logit transformation.

3.2 Data Generation of Correlated Categorical Responses

For simulation models MCOR-12 and MCOR-24 the methodology that was used to generate correlated categorical covariates was initially used by Touloumis (2016) to generate correlated nominal and binary responses under a given marginal model with categorical and/or continuous covariates, and dependence structure. We treat the corresponding simulated binary and nominal responses as covariates in our simulation study.

First we assume that $Y_{it} \in \{0,1\}$ for binary responses and $Y_{it} \in \{1,2,\ldots J \geqslant 3\}$ for multinomial responses for subject $i = 1, \ldots, n^*$ at time $t = 1, \ldots, T$ with $\mathbf{x}_{it}$ as the time varying covariate vector. Nominal responses are generated under the marginal baseline-category logit model:

$$\log\left[\frac{Pr(Y_{it}=j|\mathbf{x}_{it})}{Pr(Y_{it}=J|\mathbf{x}_{it})}\right] = (\beta_{tj0} - \beta_{tJ0}) + (\boldsymbol{\beta}_{ij} - \boldsymbol{\beta}_{tJ})^{\intercal}\mathbf{x}_{it} = \beta^*_{tj0} + \boldsymbol{\beta}^{*\intercal}_{tj}\mathbf{x}_{it}. \tag{3.1}$$

The j-th category-specific intercept at t denoted as β_{tj0} and $\boldsymbol{\beta}_{tj}$ is the j-th category-specific parameter vector corresponding with the covariates at t. The identifiability constraints are set as $\beta_{tJ0} = 0$ and $\boldsymbol{\beta}_{tJ} = \mathbf{0}$ for all t, implying $\beta^*_{tj0} = \beta_{tj0}$ and $\boldsymbol{\beta}^*_{tj} = \boldsymbol{\beta}_{tj}$ for all $t = 1, \ldots, T$ and $j = 1, \ldots, J-1$. In order to link the marginal baseline-category logit model in (3.1) with underlying regression models, the latent regression model is considered:

$$\mathbf{U}_i^{NO} = \begin{pmatrix} \mathbf{U}_{i1}^{NO} \\ \vdots \\ \mathbf{U}_{iT}^{NO} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}_{i1}^{NO} \\ \vdots \\ \boldsymbol{\mu}_{iT}^{NO} \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}_{i1}^{NO} \\ \vdots \\ \boldsymbol{e}_{iT}^{NO} \end{pmatrix} = \boldsymbol{\mu}_i^{NO} + \boldsymbol{e}_i^{NO} \tag{3.2}$$

where $\boldsymbol{U}_{it}^{NO} = (U_{it1}^{NO}, \ldots, U_{itJ}^{NO})$, $\boldsymbol{\mu}_{it}^{NO} = (\beta_{t10}\mathbf{x}_{it}, \ldots, \beta_{t(J-1)0} + \beta_{tJ}^{\intercal}\mathbf{x}_{it})^{\intercal}$ and $\boldsymbol{e}_{it}^{N0} = (e_{it1}^{NO}, \ldots, e_{itJ}^{NO})^{\intercal}$ for all i, and t, and the random vectors $\{\boldsymbol{e}_i^{NO} : i = 1, \ldots, n^*\}$ are independent if the following conditions are satisfied:

1. follows the standard extreme value distribution for all i, t and j,

2. the assumption of choice independence is met at each measurement occasion, where e_{itj}^{NO} and $e_{itj'}^{NO}$ are independent for all $j \neq j'$.

The threshold $Y_{it} = j \Leftrightarrow U_{itj}^{NO} = \max\{U_{it1}^{NO}, \ldots, U_{itJ}^{NO}\}$ is used to generate clustered nominal responses that satisfy (3.1). The association structure among the clustered nominal responses depends on the joint distribution and the correlation matrix of e_{it}^{NO} for all i, t and j. Correlated binary responses are generated under the marginal model specification:

$$Pr(Y_{it} = 1|\mathbf{x}_{it}) = F(\beta_{t0} + \boldsymbol{\beta}_t^{\intercal}\mathbf{x}_{it}), \tag{3.3}$$

where the intercept is denoted as β_{t0} at t and $\boldsymbol{\beta}_t^{\intercal}$ a vector associated with covariates at t and

F is a cdf. The latent regression model is formulated as:

$$\boldsymbol{U}_i^B = \begin{pmatrix} U_{i1}^B \\ \vdots \\ U_{iT}^B \end{pmatrix} = \begin{pmatrix} \mu_{i1}^B \\ \vdots \\ \mu_{iT}^B \end{pmatrix} + \begin{pmatrix} e_{i1}^B \\ \vdots \\ e^{iT} \end{pmatrix} = \boldsymbol{\mu}_i^B + \boldsymbol{e}_i^B \tag{3.4}$$

where $\mu_{it}^B = \boldsymbol{\beta}_t^\intercal \boldsymbol{x}_{it}$ and $\{\boldsymbol{e}_i^B : 1, \ldots, n^*\}$ are independent random vectors where $e_{it}^B \sim F$ for all i, t. Clustered binary responses are generated under these assumptions using the threshold: $Y_{it} = I(e_{it}^B \leqslant \beta_{t0} + \mu_{it}^B) = I(U_{it}^B \leqslant \beta_{t0} + 2\mu_{it}^B)$ that satisfy (3.3), and $I(A)$ is an indicator function of the event A. For each subject i, the association structure for clustered binary responses Y_{it} depends on the pairwise bivariate distributions and the correlation matrix of e_{it}^B. If e_{it}^B are independent then so are Y_{it}, for all t.

Correlated binary and nominal responses are generated via the `rbin` and `rmult.bcl` functions respectively. These functions utilizes the NORmal To Anything (NORTA) method (Cario and Nelson, 1997), which generates the continuous random vectors $\boldsymbol{U}_i$'s from the multivariate latent regression model according to the specified marginal model by specifying a desired dependence structure, which is needed prior generating responses. The NORTA method generates continuous random vectors for any type of given marginal distribution and a corresponding correlation matrix. Let F be the c.d.f of the target marginal distribution. In order to generate a p-variate random vector $\boldsymbol{W} = (W_1, \ldots, W_p)^\intercal$ with correlation matrix $\boldsymbol{R}_W$ such that $W_k \sim F$ for $k = 1, \ldots, p$, the simplified process of NORTA is described by Touloumis (2016):

1. Generate a random vector $\boldsymbol{Z} = (Z_1, \ldots, Z_p)^\intercal$ from a standard multivariate normal distribution with correlation matrix $\boldsymbol{R}_Z$, where the elements of $\boldsymbol{R}_Z$ are calculated by numerically solving $p(p-1)/2$ equations, where $\text{cor}(Z_k, Z_{k'})$ and $\text{cor}(W_k, W_{k'})$ for all $k < k'$.

2. Apply the transformation $W_k = F^{-1}[\Phi(Z_k)]$ for all k, where Φ is the cumulative distribution of the standard normal distribution.

After generating the $\boldsymbol{U}_i$'s, the correlated binary and nominal responses are generated via their respective thresholds.

Under MCOR-12 nominal responses are generated using the marginal baseline-category logit model:

$$\log\left[\frac{Pr(Y_{it}=j|\mathbf{x}_{it})}{Pr(Y_{it}=4|\mathbf{x}_{it})}\right] = \beta_{j0} + \beta_{j1}x_{i1} + \beta_{j2}x_{it2}, \tag{3.5}$$

with $T = 3$ clusters and $J = 4$ categories with $(\beta_{10}, \beta_{11}, \beta_{12}, \beta_{20}, \beta_{21}, \beta_{22}, \beta_{30}, \beta_{31}, \beta_{32}) = (1, 3, 2, 1.25, 3.25, 1.75, 0.75, 2.75, 2.25)$ and $\mathbf{x}_{it} = (x_{i1}, x_{it2})^\intercal$ with $x_{i1} \overset{iid}{\sim} N(0,1)$ and $x_{it2} \overset{iid}{\sim} N(0,1)$ for all i, t. The following correlation matrix used for NORTA is denoted as $\boldsymbol{R}^{NO}$ with elements:

$$\mathbf{R}^{NO}_{t_1j_1,t_2j_2} = \begin{cases} 1, & t_1 = t_2, j_1 = j_2 \\ 0.95, & t_1 \neq t_2, j_1 = j_2 \\ 0, & \text{otherwise} \end{cases}$$

We generate the correlated binary responses conditional on a marginal probit model:

$$\mathrm{P}(Y_{it} = 1|\mathbf{x}_{it}) = \Phi(0.2x_i) \tag{3.6}$$

with $T = 6$ clusters and $\mathbf{x}_{it} = x_i \overset{iid}{\sim} N(0,1)$ for all i, t. The intercept $\beta_{t0} = 0$ and $\beta_t = 0.2$ for all t. The following $T \times T$ correlation matrix used for the NORTA method is denoted as $\boldsymbol{R}^B$ where $r^B_{lk} = 0.75$ for $l, k = 1, \ldots, T$ such that $l \neq k$. This structure association is exchangeable with the correlation matrix for the clustered binary responses (Touloumis, 2016). The data generation of covariates corresponding to MCOR-12 are described below:

Signal Covariates

X_1, X_2, X_3 respectively correspond to the contrasts $\mathbf{c}_1$,$\mathbf{c}_2$ and $\mathbf{c}_3$, from the contrast matrix $\mathbf{C}$ for the nominal response Y_{i1}^{NO}.

X_4, X_5, X_6, respectively correspond to the binary responses Y_{i1}^{B}, Y_{i2}^{B}, and Y_{i3}^{B}.

$X_7, \ldots, X_{10} \sim MVN(\underline{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{lk})$ with $\underline{\boldsymbol{\mu}} = (0, 0, 0, 0)$, and $\sigma_{lk} = 0.8$ for $l, k = 1, 2, 3, 4$ such that $l \neq k$.

X_{11} and $X_{12} \sim N(0, 1)$.

Noise Covariates

X_{13}, X_{14} and X_{15} respectively correspond to $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ from the contrast matrix $\mathbf{C}$ of Y_{i2}^{NO}.

X_{16}, X_{17}, and X_{18} respectively correspond to the binary responses Y_{i4}^{B}, Y_{i5}^{B}, Y_{i6}^{B}.

$X_{19,} \ldots, X_{100} \sim unif(0, 1)$.

Under MCOR-24, we simulate nominal responses with $J = 4$ categories, $T = 4$ clusters, with $x_{it}^{NO} \sim N(0, 1)$ and $x_{2t}^{NO} \sim N(0, 1)$, and the binary responses with $T = 12$ clusters. We retain the same correlation structure for the NORTA method from MCOR-12 along with the latent regression coefficients. The data generation process of covariates corresponding to MCOR-24 are described below:

Signal Covariates

X_1, X_2 and X_3 respectively correspond to the contrasts $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ from the contrast matrix $\mathbf{C}$ of the response Y_{i1}^{NO}.

X_4, X_5 and X_6 respectively corresponding to the contrasts $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ from the contrast matrix $\mathbf{C}$ of the response Y_{i2}^{NO}.

X_7, X_8, X_9, X_{10}, X_{11}, X_{12} correspond to the binary responses Y_{i1}^{B}, Y_{i2}^{B}, Y_{i3}^{B}, Y_{i4}^{B}, Y_{i5}^{B}, Y_{i6}^{B} respectively.

$X_{13}, \ldots, X_{20} \sim MVN(\underline{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{lk})$, with $\underline{\boldsymbol{\mu}} = (0,0,0,0,0,0,0,0)$ and $\sigma_{lk} = 0.8$, for $l, k = 1, \ldots, 8$ such that $l \neq k$.

$X_{21}, \ldots, X_{24} \sim N(0,1)$.

Noise Covariates

X_{25}, X_{26}, and X_{27} which respectively correspond to the contrasts $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ from the contrast matrix $\mathbf{C}$ of the nominal response Y_{i3}^{NO}.

X_{28}, X_{29} and X_{30} respectively correspond to the contrasts $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ of the contrast matrix $\mathbf{C}$ from the nominal response Y_{i4}^{NO}.

$X_{31}, X_{32}, X_{33}, X_{34}, X_{35}, X_{36}$, respectively correspond to Y_{i7}^{B}, Y_{i8}^{B}, Y_{i9}^{B}, Y_{i10}^{B}, Y_{i11}^{B}, Y_{i12}^{B}.

$X_{37}, \ldots, X_{200} \sim unif(0,1)$.

For MCOR-12 and MCOR-24, all covariates that did not lie on the [0,1] domain were standardized via the inverse logit.

3.3 Performance Metrics

In order to assess the variable selection performance of each regularised OLR method across an ensemble of M fits, and each automatic threshold method, we utilize performance metrics such and true positive rate (TPR), false positive rate (FPR) and area under the ROC

curve (AUC). True positives (TP) are covariates correctly selected as signal (important) covariates with non-zero regression coefficients in the OLR model via logistic-lasso/adaptive lasso or an automatic threshold method. The TPR is represented as:

$$TPR = \frac{TP}{r} \tag{3.7}$$

where r corresponds to the number of signal covariates in the OLR model. The false positives (FP) are those covariates corresponding to zero valued coefficients in the OLR model (noise) that are incorrectly selected as signal, via the logistic-lasso/adaptive lasso or an automatic threshold method. The FPR is represented as:

$$FPR = \frac{FP}{p - r} \tag{3.8}$$

where p and r correspond to the total number of covariates and number of covariates with non-zero regression coefficients in the OLR model respectively. The AUC score is used to evaluate the overall variable selection performance, where a large AUC represents a good measure between the TPR and FPR, indicating a higher power in selecting signal covariates (Guo et al., 2015). The AUC score approximation is represented by López et al. (2015) as:

$$AUC = \frac{1 + TPR - FPR}{2}. \tag{3.9}$$

These performance metrics are the criteria by which variable selection performance of our algorithm is examined.

Chapter 4

Simulation Results

In this chapter we examine our algorithm under the four simulation models; MUNCOR-12, MUNCOR-24, MCOR-12, and MCOR-24 with three imbalance ratios I_R = 1:50, 1:100, and 1:1000 across the balanced sample sizes n_b = 1000, 2000, 3000, 4000, and 5000. The model fitting process of our algorithm considers an ensemble of $M = 500$ model fits of the lasso, adaptive lasso and ridge regression selected with "lambda.min". In section 4.4, we observe the effect of λ on variable selection. We consider the lasso, adaptive lasso, and ridge regression models selected with "lambda.1se", under MCOR-24 when I_R = 1:1000. Within the simulation study, we examine ranked based automatic variable selection methods for various regularised OLR models and assess their performance via the TPR, FPR and AUC scores.

The automatic variable selection methods under examination are the maximum rank difference, mean changepoint detection, and complete linkage clustering methods, and are applied to the $Rank(x_i)$ and/or P_{Drop} values obtained from the ensemble of regularised OLR model fits. The performance metric scores corresponding to each automatic variable selection method are denoted as:

mRank: When the maximum difference method is applied to $Rank(x_i)$;

mPdrop: When the maximum difference method is applied to P_{Drop} values;

cRank: When the mean changepoint detection method is applied to $Rank(x_i)$;

cPdrop: When the mean changepoint detection method is applied to P_{Drop} values.

For ridge regression we applied the mean changepoint detection method on the sorted absolute valued standardized regression coefficients for each individual model fit (across $M = 500$). This is done to observe whether automatic variable selection on ridge regression estimates can attain stable variable ranks across an ensemble of fits, in comparison to l_1 regularisation methods which are usually implemented instead. In contrast to this, we also computed $Rank(x_i)$ scores based on the ensemble of ridge regression estimates and utilized the maximum difference and mean changepoint methods to discern whether or not reasonable results can be achieved without automatic variable selection across each individual model fit (i.e using l_1 regularisation or mean change point detection).

4.1 Variable Ranking and Selection Performance with Un-Correlated Data

4.1.1 MUNCOR-12

We examine simulation model MUNCOR-12 ($p = 100$), which is one of the baseline simulation models. In Table A3 the mean TPR scores over the ensemble of model fits across all sample sizes range from at least 0.900 to 1.000 for both the lasso and adaptive lasso. The TPR scores corresponding to ridge regression range from 0.840 to 0.990. The automatic selection methods tend to recover most to all signal covariates across all sample sizes when applied to $Rank(x_i)$ and or P_{Drop} values for the lasso and adaptive lasso fits (Table A3). The adaptive lasso does not out perform the lasso regarding recovering the signal covariates across

the ensemble. When the mean change point detection method is applied to each individual ridge regression fit (cRank), most signal covariates are recovered on average across all sample sizes (Table A3). The effect of sample size on the distribution of the individual TPR scores is similar for all regularisation methods, except for ridge regression showing evident variability at $n_b = 1000$ (Figure 4.1). In Figure 4.1, cRank appears to recover all signal covariates across most sample sizes.

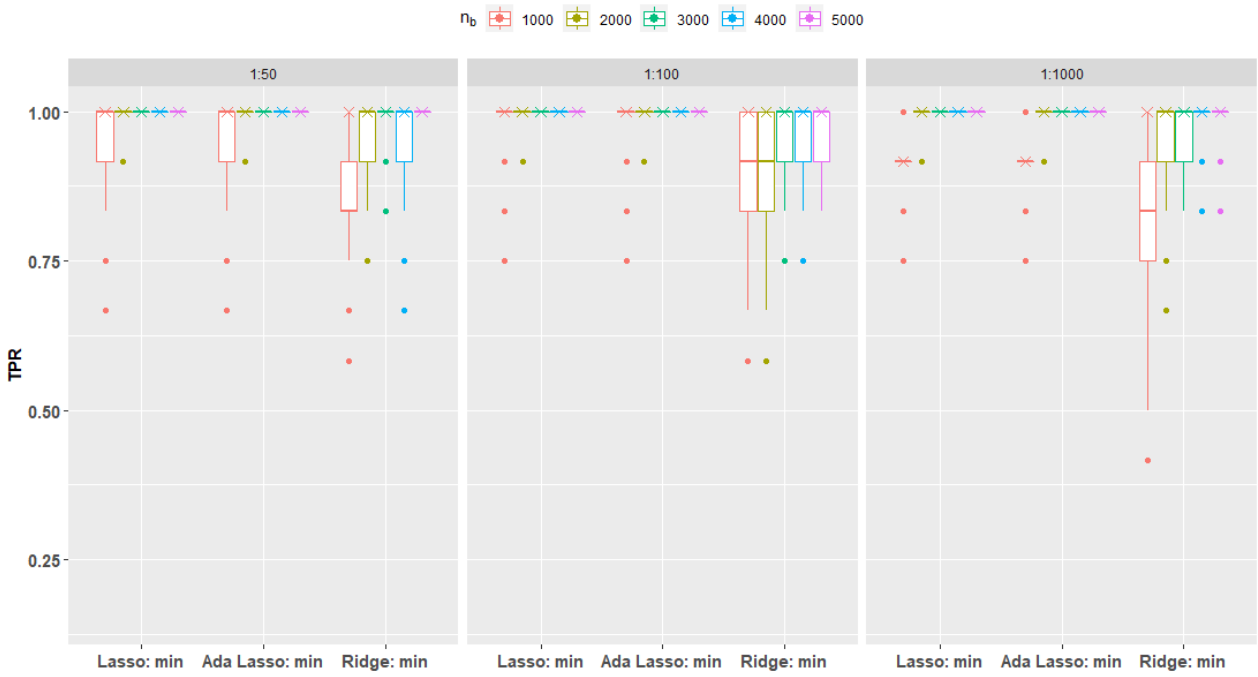


Figure 4.1: MUNCOR-12 - Distribution of variable selection TPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "×" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

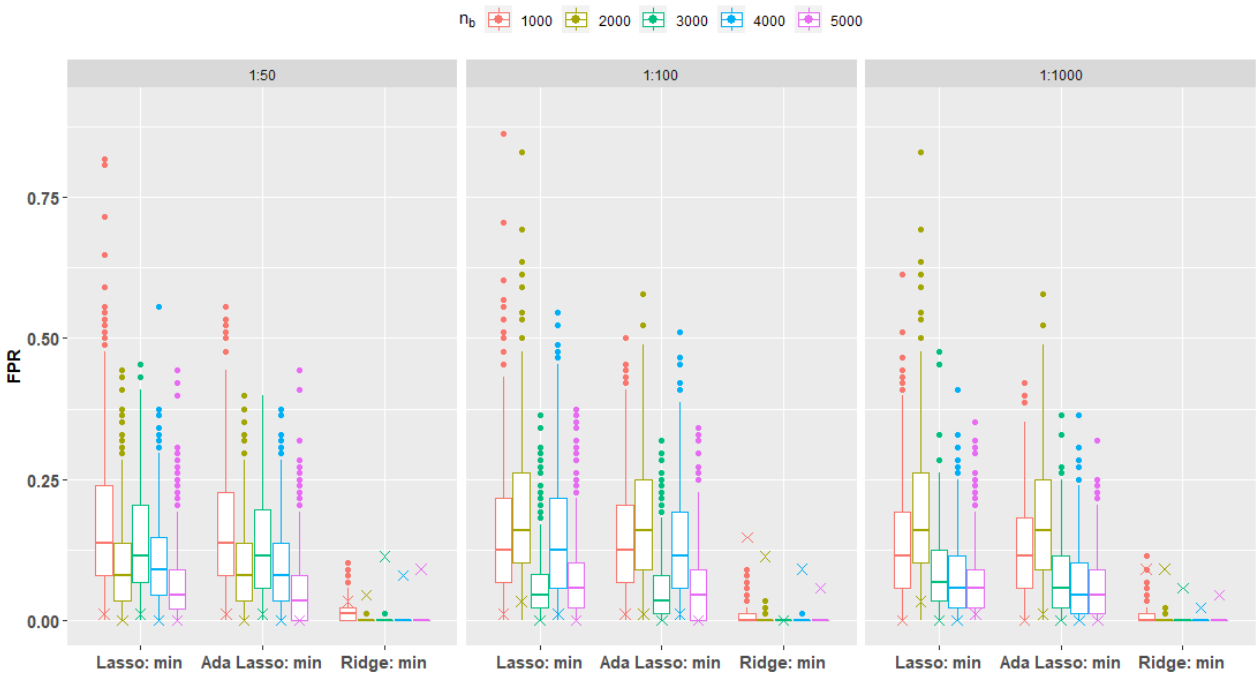


Figure 4.2: MUNCOR-12 - Distribution of variable selection FPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

On the other hand, several noise covariates were selected on average across the ensemble of the lasso and adaptive lasso fits. The highest mean FPR score across the ensemble for the lasso and adaptive lasso models was 0.191 and 0.177 respectively, which is approximately 17 and 16 noise covariates selected on average (Table A6). Across all sample sizes the adaptive lasso achieved lower mean FPR score than the lasso, whereas ridge regression out performed both the lasso and adaptive lasso in terms of mean FPR (Table A6). All automatic selection methods either filter all or most noise covariates, and we see that there is considerable variation amongst the FPR scores for the lasso and adaptive models, but the median score generally drops with increasing sample size for all imbalance ratios (Table A6; Figure 4.2). There are also many potential outliers in Figure 4.2 under the lasso and adaptive

lasso models, across all imbalance ratios. In Figure 4.2 there is a lasso model that had selected around 90% of all noise covariates (when I_R = 1:100), highlighting the importance of generating an ensemble of model fits with aggregate variable ranking to stabilize rank scores. In Figure 4.2, cRank tends to select several noise covariates for ridge regression, more than at least 75% of the individual fits. We see that cRank with ridge regression ranks attained a similar FPR score to the mean/median FPR scores of the lasso and adaptive lasso, for example the highest FPR score corresponding to cRank is 0.091 (when I_R = 1:1000) approximately 8 noise covariates out of 88 (Table A6; Figure 4.2).

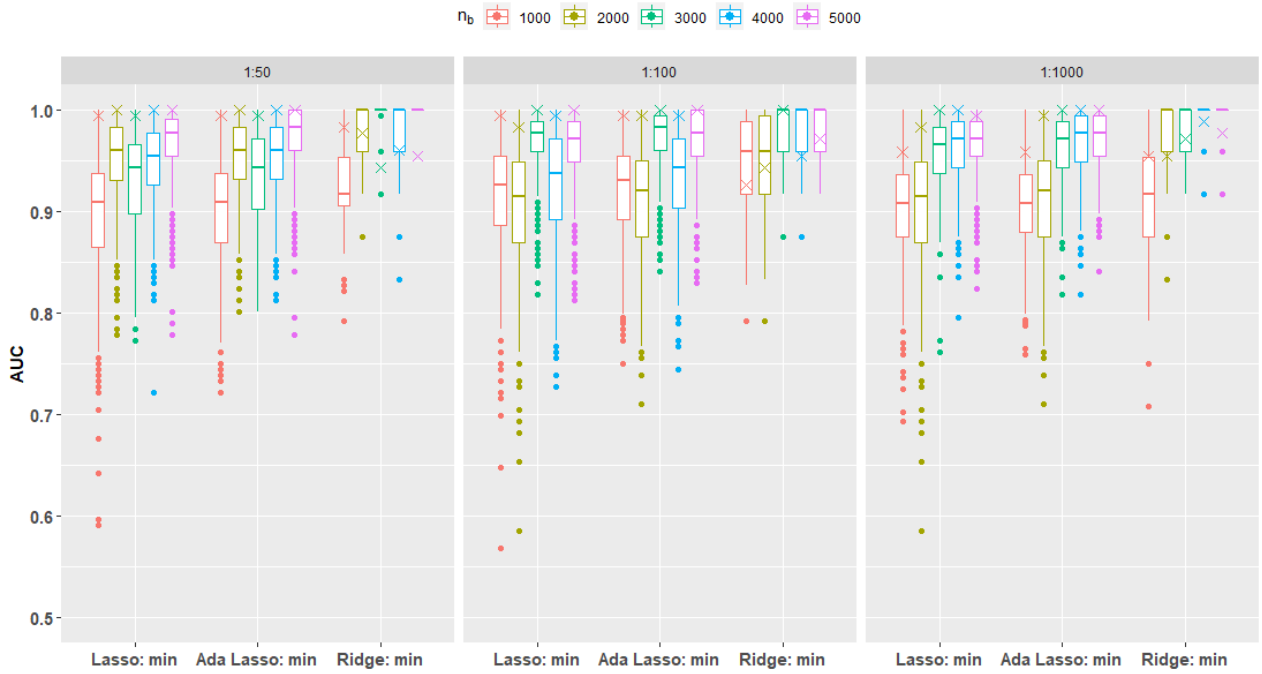


Figure 4.3: MUNCOR-12 - Distribution of variable selection AUC scores across M = 500 model fits for each n_b across all I_R's. The marker "×" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

We conclude from the AUC scores reported in Table 4.1 and Figure 4.3 that; i) each regularised OLR method generated stable variable rankings, ii) the adaptive lasso is only

a slight improvement upon the standard lasso across the ensemble of fits, iii) automatic selection is insensitive to severe class imbalance. In Figure 4.3, the AUC scores corresponding to cRank are higher than majority of the individual model fits, scoring well above the 75th percentile for all samples sizes corresponding for the lasso and adaptive lasso models. The AUC scores corresponding to cRank with ridge regression tended to score below the median AUC, although its respective score was still above 0.900 (Figure 4.3).

Table 4.1: MUNCOR-12 - Variable selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.898 (0.050)	0.958	0.958	0.958	0.947	0.958
	2000	0.903 (0.062)	1.000	0.983	1.000	0.983	0.983
	3000	0.957 (0.036)	1.000	1.000	1.000	1.000	0.972
	4000	0.961 (0.034)	1.000	1.000	1.000	1.000	1.000
	5000	0.966 (0.030)	1.000	0.994	0.994	0.994	0.994
Ada Lasso: min	1000	0.903 (0.043)	0.958	0.958	0.958	0.947	0.93
	2000	0.910 (0.054)	1.000	0.994	1.000	0.983	0.983
	3000	0.960 (0.035)	1.000	1.000	1.000	1.000	1.000
	4000	0.966 (0.034)	1.000	1.000	1.000	1.000	1.000
	5000	0.972 (0.028)	1.000	1.000	1.000	0.994	0.994
Ridge: min	1000	0.918 (0.043)	0.958	0.955	-	-	-
	2000	0.984 (0.028)	0.958	0.955	-	-	-
	3000	0.985 (0.021)	1.000	0.972	-	-	-
	4000	0.995 (0.014)	1.000	0.989	-	-	-
	5000	0.992 (0.017)	0.994	0.977	-	-	-

4.1.2 MUNCOR-24

Here we examine the variable selection performance under MUNCOR-24 ($p = 200$). It is clear that a lower proportion of signal covariates were recovered across all regularised OLR model ensembles in comparison to MUNCOR-12, however the mean TPR increased as sample size increased (Table A9; Figure 4.4). Although the automatic threshold methods did not recover a high proportion of signal covariates relative to MUNCOR-12, we see that cRank managed to recover more signal covariates than approximately 25-50% of the lasso/adaptive lasso fits and much more than 75% of them ridge regression fits respectively (Table A9; Figure 4.4). Overall, all automatic selection methods perform relatively the same in terms of recovering signal covariates (Table A9).

Figure 4.4: MUNCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "×" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

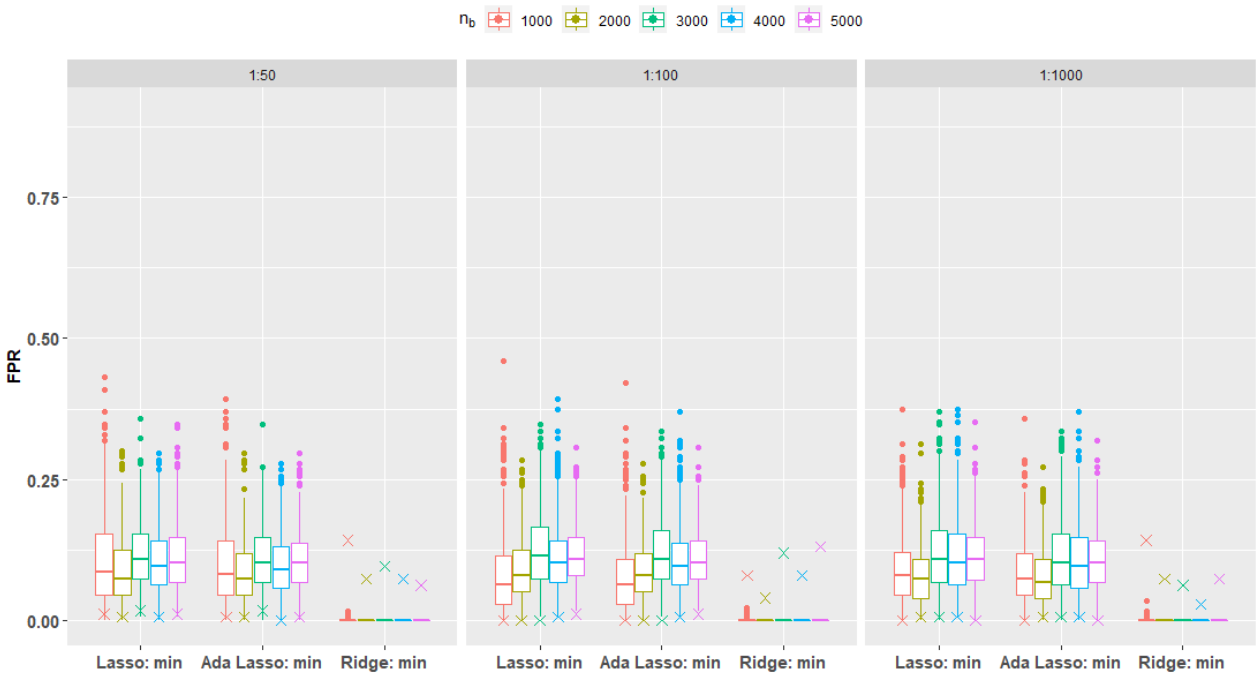


Figure 4.5: MUNCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

The mean FPR scores corresponding to the lasso and adaptive lasso are only lower at $n_b = 1000, 2000$ relative to the mean FPR scores for the lasso and adaptive lasso in MUNCOR-12 (Table A12). Ridge regression under MUNCOR-12 and MUNCOR-24 attained similar mean FPR scores, screening out majority of the noise covariates (Table A6; Table A12). Although the proportion of signal covariates recovered by the automatic threshold methods were not relatively as high as under MUNCOR-12, each method still screened out the majority of the noise covariates (Table A12). The highest mean FPR score attained by the lasso and adaptive lasso models are 0.116 and 0.115 respectively, which is approximately 20 noise covariates selected on average out of 176 (Table A12). In Figure 4.5, the individual FPR scores have similar distributions between sample sizes and across imbalance ratios and

do not decrease with increasing sample sizes along with the mean/median FPR. It is evident that cRank (and other threshold methods) screens out the majority of the noise covariates compared to the individual model fits regarding the lasso and adaptive lasso, however cRank with ridge regression rankings select a substantial amount of noise (Table A12; Figure 4.2). In Table A12 and Figure 4.2, we see cRank selects an amount of noise covariates similar to the mean/median FPR scores for the lasso and adaptive lasso, where the highest cRank FPR score (when $I_R = 1{:}1000$) is 0.142 which is approximately 25 noise covariates.

In Table 4.2, the mean AUC scores between the ensemble of lasso and adaptive lasso model fits range from 0.860-0.900, whereas under MUNCOR-12 the respective mean AUC scores range from 0.950-1.000 (Table 4.1). The mean AUC scores across the ensemble of ridge regression fits score around 0.700 whereas under MUNCOR-12, the respective mean AUC scores are above 0.950 (Table 4.2; Table 4.1). The mean AUC across the ensemble of adaptive lasso fits are a minor improvement upon the corresponding mean AUC for the lasso, and the automatic selection methods attained very similar performances for both lasso and adaptive lasso (Table 4.2). The AUC scores corresponding to the automatic threshold methods are reasonable and generally improve as sample size increases ranging from around 0.730-0.940 between all regularised methods (Table 4.2). Only the AUC scores corresponding to cRank with ridge regression score above 0.900 across all sample sizes (Table 4.4; Figure 4.6). In Figure 4.4, the AUC scores corresponding to cRank for all regularised methods often perform better than 75% of the AUC scores attained with an individual model fit for most sample sizes.

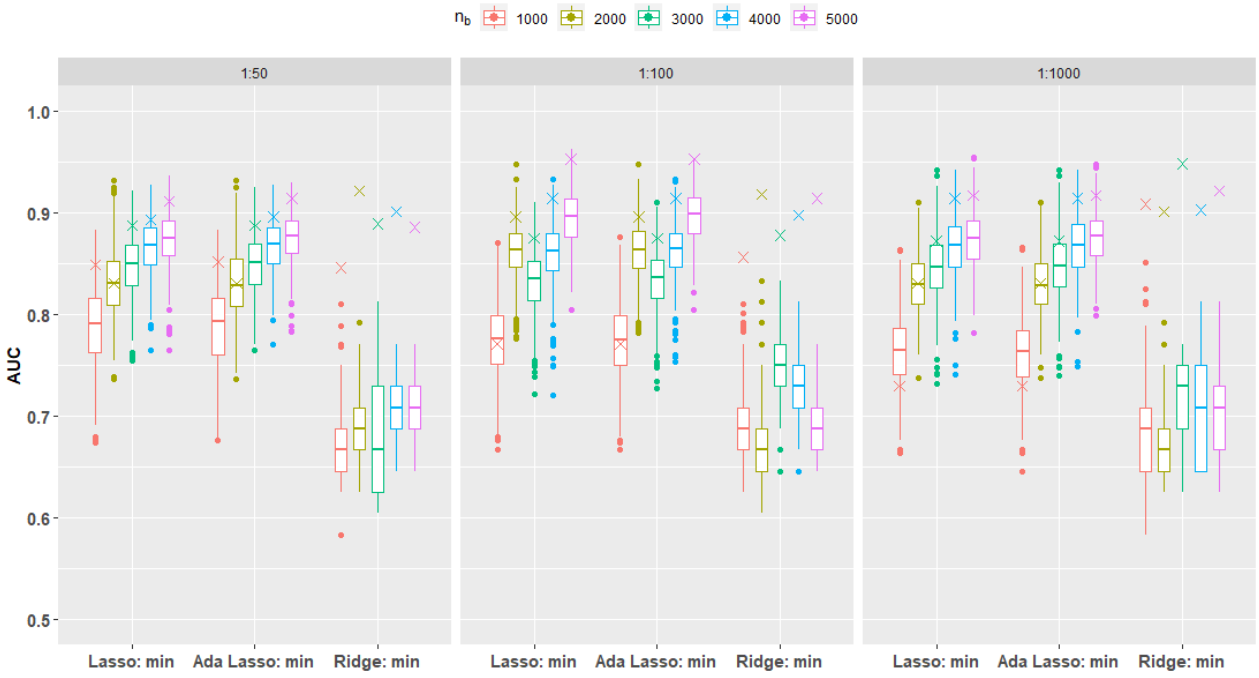


Figure 4.6: MUNCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Table 4.2: MUNCOR-24 - Variable selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.763 (0.035)	0.729	0.729	0.729	0.786	0.729
	2000	0.830 (0.028)	0.833	0.830	0.833	0.848	0.848
	3000	0.846 (0.033)	0.833	0.872	0.872	0.914	0.833
	4000	0.865 (0.030)	0.908	0.914	0.908	0.908	0.908
	5000	0.874 (0.028)	0.917	0.917	0.917	0.914	0.917
Ada Lasso: min	1000	0.762 (0.035)	0.729	0.729	0.729	0.786	0.729
	2000	0.830 (0.028)	0.833	0.830	0.833	0.848	0.848
	3000	0.848 (0.031)	0.833	0.872	0.833	0.914	0.872
	4000	0.866 (0.030)	0.875	0.914	0.908	0.908	0.908
	5000	0.875 (0.027)	0.917	0.917	0.917	0.914	0.917
Ridge: min	1000	0.680 (0.047)	0.771	0.908	-	-	-
	2000	0.674 (0.040)	0.854	0.901	-	-	-
	3000	0.717 (0.033)	0.896	0.948	-	-	-
	4000	0.704 (0.047)	0.914	0.902	-	-	-
	5000	0.699 (0.034)	0.917	0.921	-	-	-

4.2 Variable Ranking and Selection Performance with Correlated Data

4.2.1 MCOR-12

We examine the first simulation model that possesses correlation between a subset of signal and noise covariates, with correlation for nominal covariates:

$$\mathbf{R}^{NO}_{t_1 j_1, t_2 j_2} = \begin{cases} 1, & t_1 = t_2, j_1 = j_2 \\ 0.95, & t_1 \neq t_2, j_1 = j_2 \\ 0, & \text{otherwise} \end{cases}$$

and correlation matrix corresponding to binary covariates with entries; $r^B_{lk} = 0.75$ for $l, K = 1, \ldots, T$ such that $T \neq K$. In Table A15 the mean TPR scores across the ensemble of each regularised models were very similar to the mean TPR scores attained under MUNCOR-12, scoring around 0.900-1.000 (Table A3) but are slightly lower on average. Likewise, all automatic selection methods tend to recover most or all signal covariates despite the severe class imbalance and high correlation between the signal and noise covariates and attained similar TPR scores (Table A15). The TPR scores corresponding to the automatic selection methods and the mean TPR scores corresponding to the regularised methods often recovered more signal covariates than the respective scores under MUNCOR-24 (Table A15; Table A9). In Figure 4.7, the distribution of TPR scores for each regularised model exhibited high variation at $n_b = 1000$ and $n_b = 2000$, and was often the case across all sample sizes for ridge regression. The variation of the distributed TPR scores displayed less variation and the median TPR scores corresponding to all regularised models score above 0.750.

In Table A18, the mean FPR for the adaptive lasso is lower than the mean FPR scores

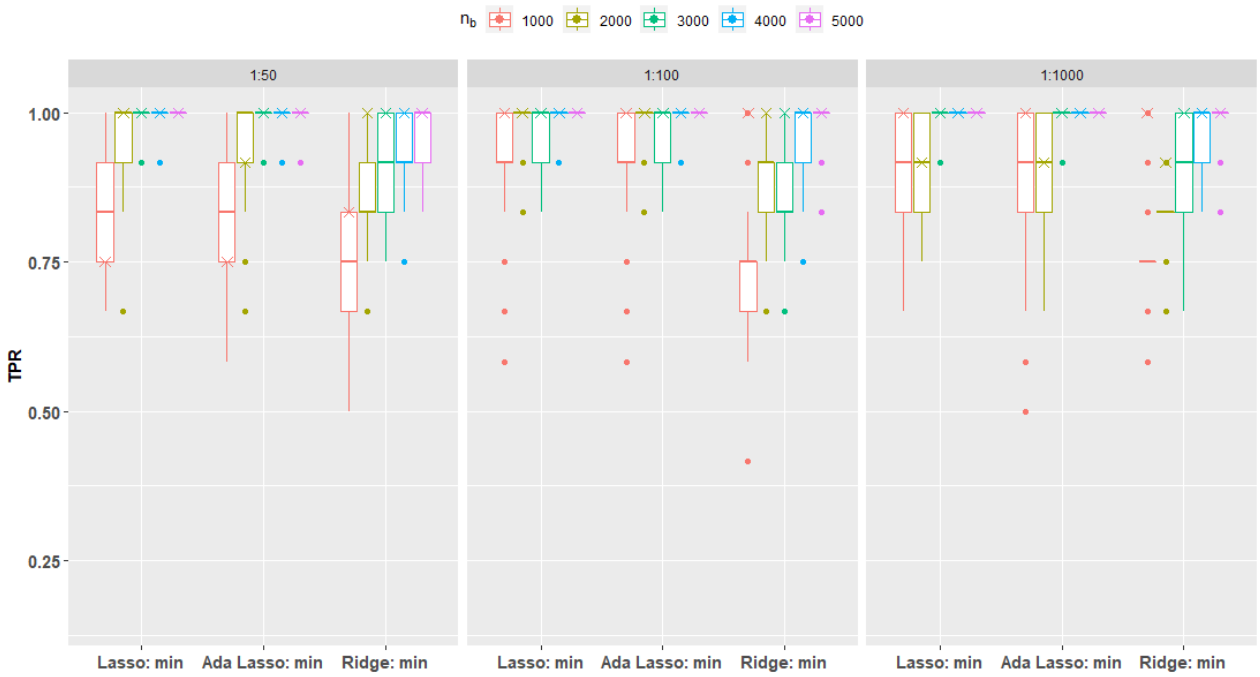


Figure 4.7: MCOR-12 - Distribution of variable selection TPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

for the lasso across all sample sizes, whereas ridge regression screened out most noise covariates across the ensemble, exhibiting similar mean FPR scores under MUNCOR-12 and MUNCOR-24. The highest mean FPR score corresponding to the lasso and adaptive lasso is and 0.243 and 0.205 respectively, which is approximately 21 and 18 noise covariates selected on average, compared to 17 and 16 under MUNCOR-12. In Table A18, all automatic selection methods tend to filter most to all noise covariates, with similar FPR scores under MUNCOR-12 and MUNCOR-24. In Figure 4.8, the median FPR scores do not vary as much between samples sizes relative to the median FPR scores under MUNCOR-12, and generally do not decrease as sample size increases but for ridge regression (Figure 4.2). The cRank FPR scores corresponding to ridge regression attains similar scores to the mean/median lasso

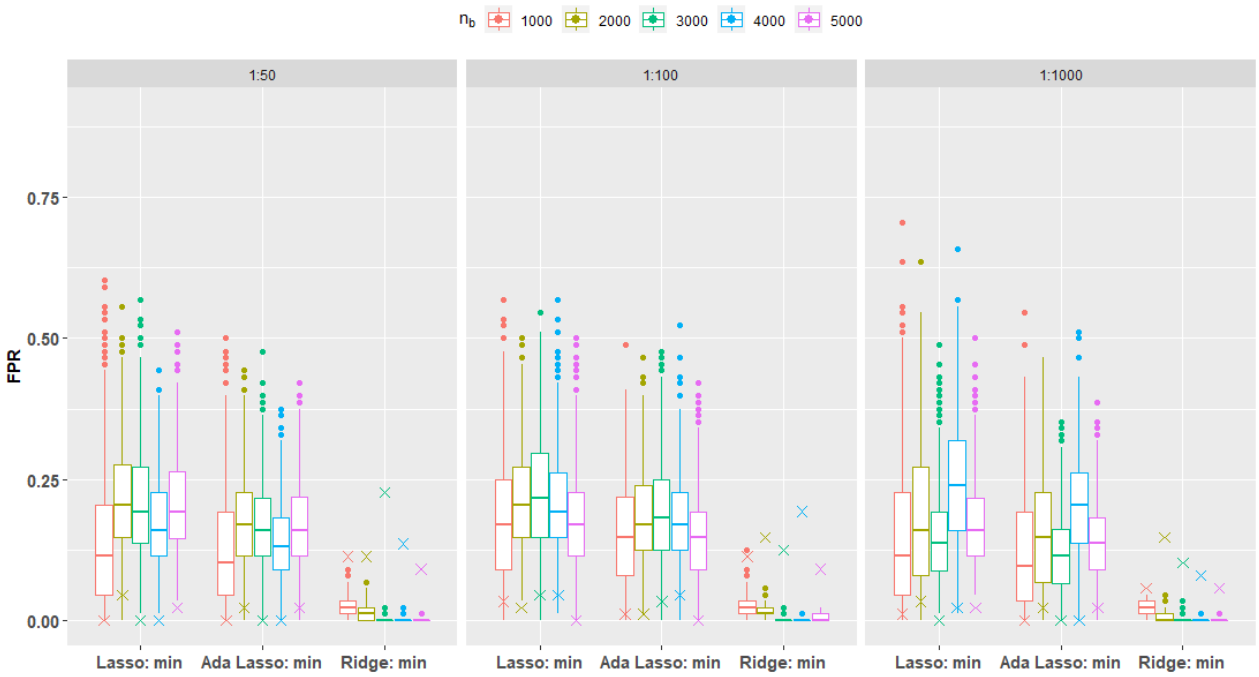


Figure 4.8: MCOR-12 - Distribution of variable selection FPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "×" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

and adaptive lasso FPR scores, for example the highest cRank FPR score is 0.148 which is approximately 13 noise covariates, compared to 8 under MUNCOR-12 (Table A18; Figure 4.8)

In Table 4.3, the mean AUC scores for the lasso and adaptive lasso range between around 0.900-0.930, and are lower than the mean AUC scores for the lasso and adaptive lasso under MUNCOR-12 and higher than the respective AUC scores under MUNCOR-24. The automatic threshold methods often obtain an AUC close 1.000 for all regularised models, and are quite similar to the respective AUC scores under MUNCOR-12 (Table 4.3; Table 4.1). In Table 4.3, the same trend in MUNCOR-12 and MUNCOR-24 appears, where the adaptive lasso is only a minor improvement upon the lasso according to the mean AUC scores. In

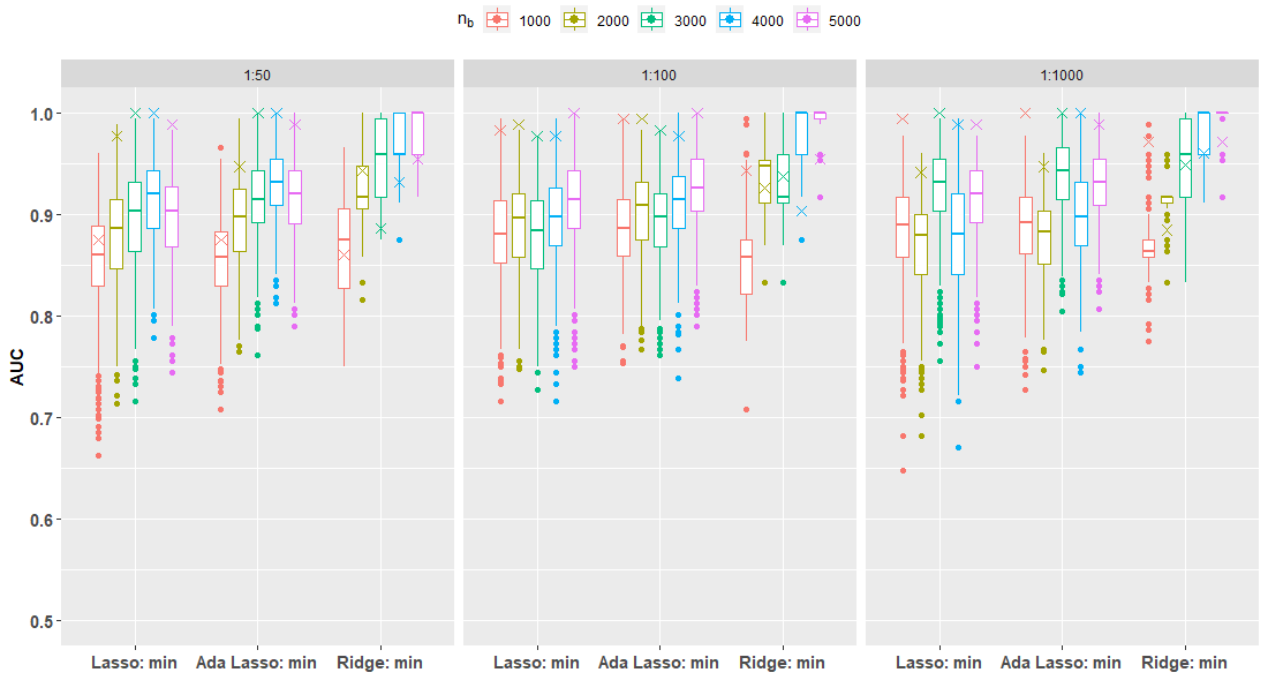


Figure 4.9: MCOR-12 - Distribution of variable selection AUC scores across = 500 model fits for each n_b across all I_R's. The marker "×" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Figure 4.9, the distributed AUC scores do not exhibit major variation between sample size and the median AUC scores generally increase as sample size increases. The AUC scores corresponding to cRank often score well above the 75th percentile, however we see this less often with ridge regression likewise the same trend appears under MUNCOR-12, however, the cRank AUC scores with ridge regression often score above 0.900 (Table 4.3; Figure 4.9). It is clear that variable ranking algorithm is robust to severe class imbalance and correlation between some signal and noise covariates under MCOR-12 (Figure 4.9; Table 4.3).

Table 4.3: MCOR-12 - Variable selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.883 (0.049)	0.994	0.994	0.994	0.994	0.994
	2000	0.868 (0.045)	0.917	0.941	0.917	0.907	0.917
	3000	0.925 (0.044)	1.000	1.000	1.000	0.994	1.000
	4000	0.879 (0.055)	1.000	0.989	1.000	0.977	0.994
	5000	0.913 (0.040)	1.000	0.989	1.000	0.977	0.989
Ada Lasso: min	1000	0.886 (0.043)	0.994	1.000	0.994	0.994	0.833
	2000	0.876 (0.038)	0.917	0.947	0.917	0.924	0.917
	3000	0.937 (0.039)	1.000	1.000	1.000	1.000	1.000
	4000	0.898 (0.046)	1.000	1.000	1.000	0.977	1.000
	5000	0.929 (0.035)	1.000	0.989	1.000	0.983	0.972
Ridge: min	1000	0.868 (0.029)	0.900	0.972	-	-	-
	2000	0.910 (0.017)	0.905	0.884	-	-	-
	3000	0.956 (0.036)	0.994	0.949	-	-	-
	4000	0.981 (0.029)	0.994	0.960	-	-	-
	5000	0.997 (0.008)	0.977	0.972	-	-	-

4.2.2 MCOR-24

The mean TPR scores for lasso/adaptive lasso and ridge regression range around 0.900-1.000 and 0.750-1.000 respectively (Table A21). The mean TPR scores in Table A21 corresponding to all regularised models are higher than their respective mean TPR scores under MUNCOR-24, but lower than the respective mean TPR scores under MUNCOR-12 and MCOR-12. We see that each automatic selection method in Table A21 performs quite well at most sample sizes, however there is evident variation among the TPR scores corresponding the threshold methods. For example, at $n_b = 1000$ the adaptive lasso the TPR score is 0.292, and at $n_b = 3000$ the respective TPR score is 0.958 (Table A21). Likewise, the

TPR score corresponding to clustering with the adaptive lasso is 0.542 at $n_b = 1000$ and 0.958 at $n_b = 3000$ (Table A21). We see that all signal covariates are recovered with cRank corresponding to ridge regression for all sample sizes, this is also the case with cRank and ridge regression under MUNCOR-12 (when $I_R = 1{:}1000$) (Table A21; Table A3). In Figure 4.10 there appears to be high variation among the distributed TPR scores between sample sizes, likewise across the imbalance ratios. The TPR scores corresponding to cRank with the lasso and adaptive lasso tend to score below the median TPR, however recovered at least 75% of the signal covariates (Figure 4.10). Despite the evident variation among the distributed TPR scores with ridge regression, automatic selection corresponding to cRank managed to recover most signal covariates (Figure 4.10).

Figure 4.10: MCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "×" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

In Table A24 the mean FPR scores for the lasso and adaptive lasso, are lower compared with the respective mean FPR scores under MCOR-12, although they are quite similar to the corresponding values under MUNCOR-12 and MUNCOR-24 (Table A6; Table A12; Table A18). In Table A24 the mean FPR with ridge regression is higher than the respective mean FPR scores under the other simulation models. The FPR scores attained with the automatic selection methods are quite similar between each other and to the corresponding FPR values under the rest of the simulation models, screening out most noise covariates (Table A24). The highest mean FPR scores corresponding to the lasso and adaptive lasso are 0.113 and 0.100 which is approximately 20 and 18 noise covariates selected on average respectively, compared to 20 noise covariates selected by the lasso and adaptive lasso under MUNCOR-24. Likewise with the other simulation models, cRank with ridge regression selects more noise covariates than the majority ridge regression models when cRank was applied to each individual fits (Figure 4.11). The highest FPR score attained with cRank for ridge regression (when $I_R =$ 1:1000) is 0.148 which is approximately 26 noise covariates, compared to 25 under MCOR-12.

In Table 4.4 the mean AUC scores for the lasso, and adaptive lasso are similar to the respective AUC scores under MUNCOR-24 and MCOR-12, and range from 0.889-0.920 (Table 4.2; Table 4.3). Overall, there are not any major differences regarding the mean AUC scores for the lasso and adaptive lasso compared to the respective mean AUC scores under the other simulation models (Table 4.1; Table 4.2; Table 4.3; Table 4.4). The mean AUC scores corresponding to ridge regression under MCOR-24 do not score above 0.900, however they are only higher than the respective AUC scores under MUNCOR-24 (Table 4.4; 4.2). Each automatic selection method usually scored at least above 0.800, and under some sample sizes above 0.900 (Table 4.4). The AUC scores attained by each automatic threshold method are similar overall, although the mRank AUC score corresponding to adaptive lasso is 0.646 at $n_b = 1000$ (Table 4.4). The cRank AUC scores tend to place above the median AUCs for most sample sizes for all regularised models (Figure 4.12).

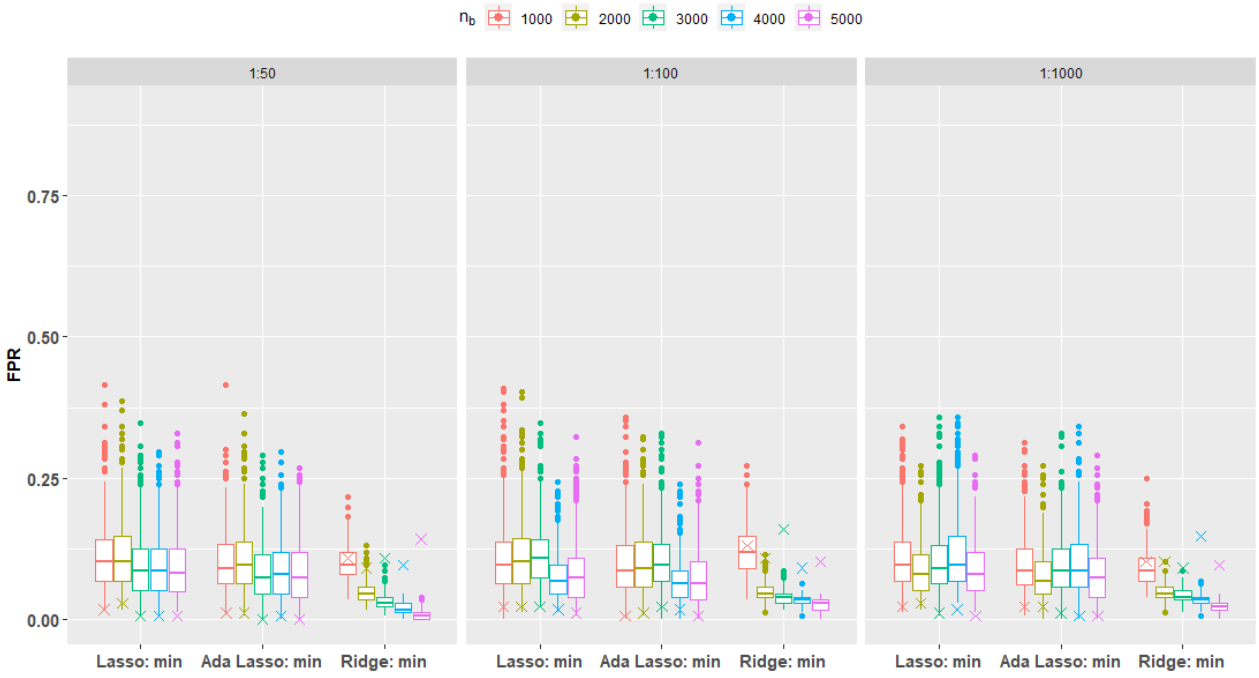


Figure 4.11: MCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

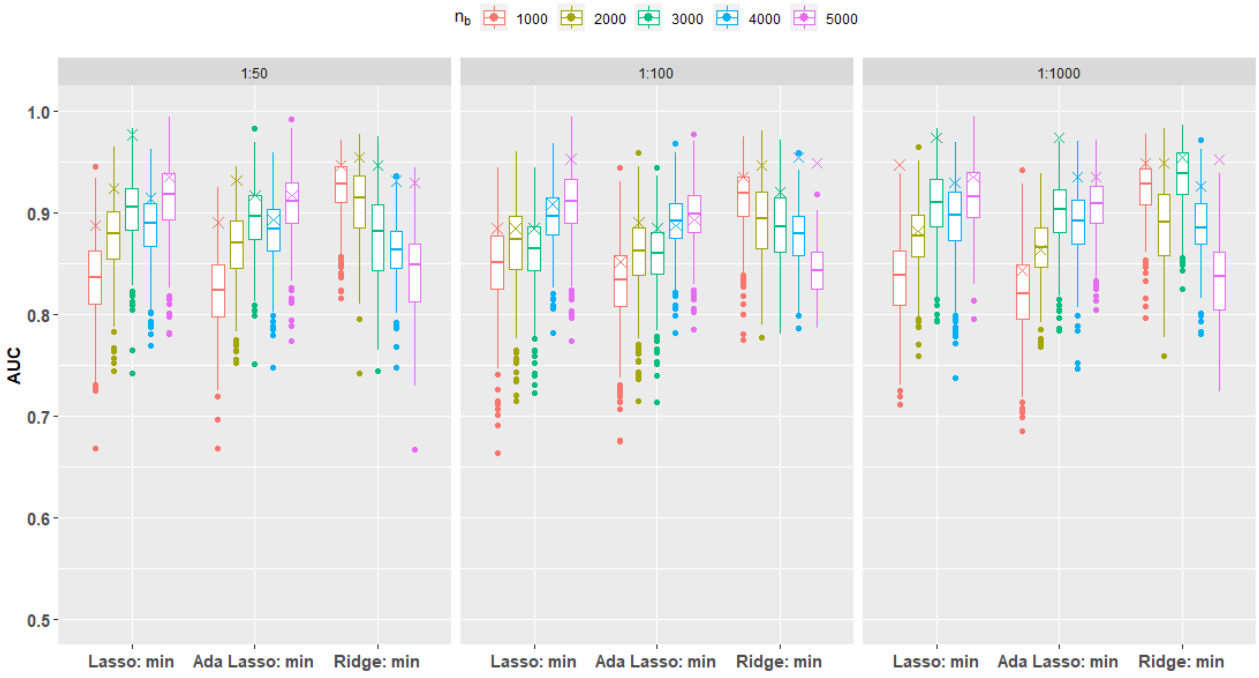


Figure 4.12: MCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits for each n_b across all I_R's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Table 4.4: MCOR-24 - Variable selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.895 (0.028)	0.830	0.947	0.830	0.933	0.830
	2000	0.893 (0.037)	0.882	0.882	0.882	0.882	0.882
	3000	0.913 (0.035)	0.973	0.973	0.973	0.971	0.973
	4000	0.909 (0.033)	0.935	0.929	0.929	0.941	0.929
	5000	0.915 (0.033)	0.935	0.935	0.935	0.947	0.977
Ada Lasso: min	1000	0.891 (0.027)	0.646	0.843	0.848	0.936	0.771
	2000	0.888 (0.033)	0.864	0.864	0.864	0.884	0.882
	3000	0.907 (0.032)	0.973	0.973	0.973	0.973	0.973
	4000	0.898 (0.030)	0.893	0.935	0.929	0.950	0.929
	5000	0.906 (0.029)	0.935	0.935	0.935	0.935	0.935
Ridge: min	1000	0.877 (0.027)	0.956	0.949	-	-	-
	2000	0.887 (0.033)	0.956	0.949	-	-	-
	3000	0.841 (0.039)	0.972	0.955	-	-	-
	4000	0.846 (0.024)	0.949	0.926	-	-	-
	5000	0.834 (0.038)	0.941	0.952	-	-	-

4.3 The Effect of Penalization Parameter λ

Here we study the effect that the shrinkage parameter λ has on variable ranking and selection demonstrated under MCOR-24. In Section 4, we studied variable ranking and selection on regularised models selected at $\lambda =$ "lambda.min", the value that minimizes the mean cross-validation error during the model fitting process. We now consider the value of λ that is one standard error away from "lambda.min", that obtains a more parsimonious model for l_1 regularisation.

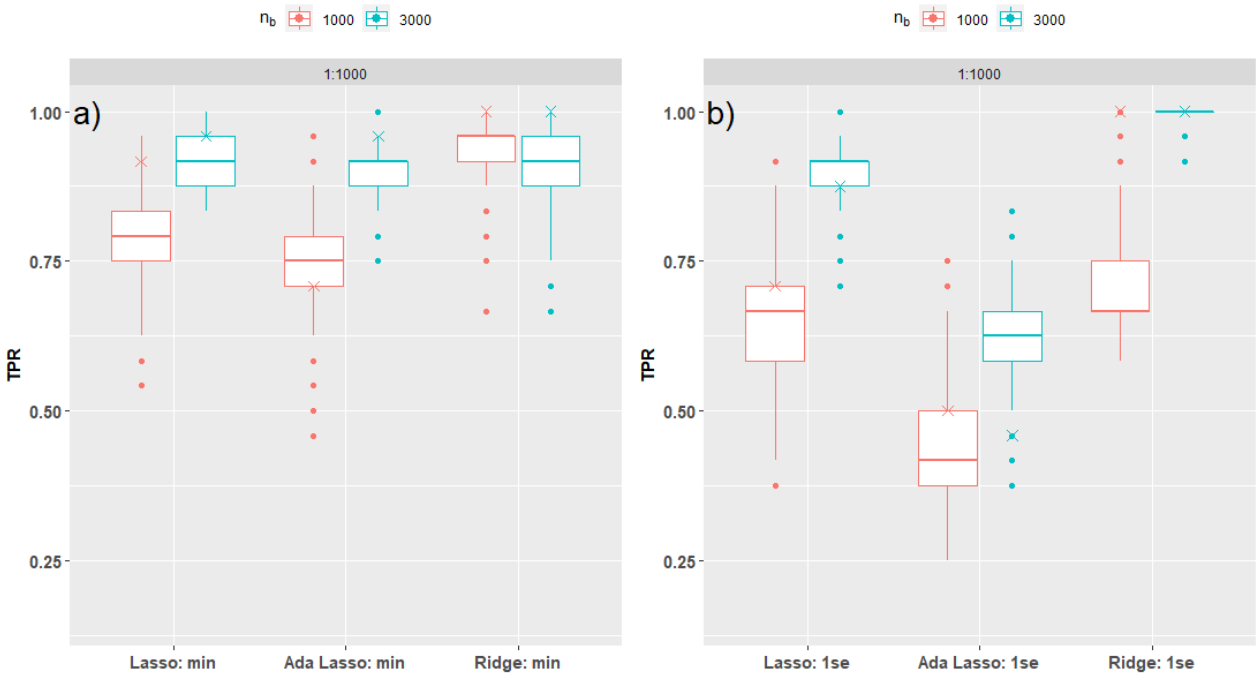


Figure 4.13: MCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits with (a) $\lambda =$ "lambda.min" and (b) $\lambda =$ "lambda.1se", for $n_b = 1000, 3000$ and $I_R =$ 1:1000. The marker "$\times$" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

In Table A33 and Figure 4.13, it is evident that λ at "lambda.1se" penalizes the lasso and adaptive lasso coefficients much to the extent where a high proportion of signal covariates

are zeroed-out across the $M = 500$ fits. Due to this effect, the resultant variable rankings obtained from the ensemble of fits are unstable, leading to the automatic selection methods failing to recover the signal covariates since the variable rankings do not show a clear distinction between the set of signal and noise covariates. In Table A33, the mean TPR values range from approximately 0.660-0.900 and 0.440-0.680 for the lasso and adaptive lasso respectively, whereas for ridge regression, the mean TPR scores are close to 1.000 except at $n_b = 1000$ (Table A33). The mean TPR scores attained with the lasso at "lambda.1se" are lower than than the mean TPR scores with "lambda.min" but not substantially, however this is not the case for the adaptive lasso (Table A21; Table A33). The automatic selection methods with the adaptive lasso obtained a TPR score between approximately 0.040-0.300, and observe similar scores with the other automatic selection methods, clearly indicating unstable variable rankings (Table A33). The cRank and mRank TPR scores with ridge regression recover all signal covariates for each sample size (Table A33). In Figure 4.13, the distributed TPR scores of with "lambda.min" exhibit similar trends to the distribution of TPR scores with "lambda.1se" as we tend to see TPR increasing with sample size, and more variation among the latter. We see that cRank recovers a higher proportion of signal covariates at $n_b = 1000$ relative to $n_b = 3000$ and "lambda.1se" for the lasso and adaptive lasso, scoring no higher than the 25% of the individual model fits (Figure 4.13).

In Table A34, the mean of FPR scores for all regularised models are much lower, where ridge regression attained a score as high as 0.066 (approximately 12 noise covariates selected on average) and hence majority of the noise covariates were screened out by the automatic selection methods. In Figure 4.14, the distributed FPR scores corresponding to "lambda.1se" depict less variation and potential outliers compared to the respective FPR distribution with "lambda.min", and the median FPR scores tend to place below 0.05. We also notice the similar trend observed in each simulation model at "lambda.min" where cRank scores a higher FPR than the majority of the individual model fits corresponding to ridge regression

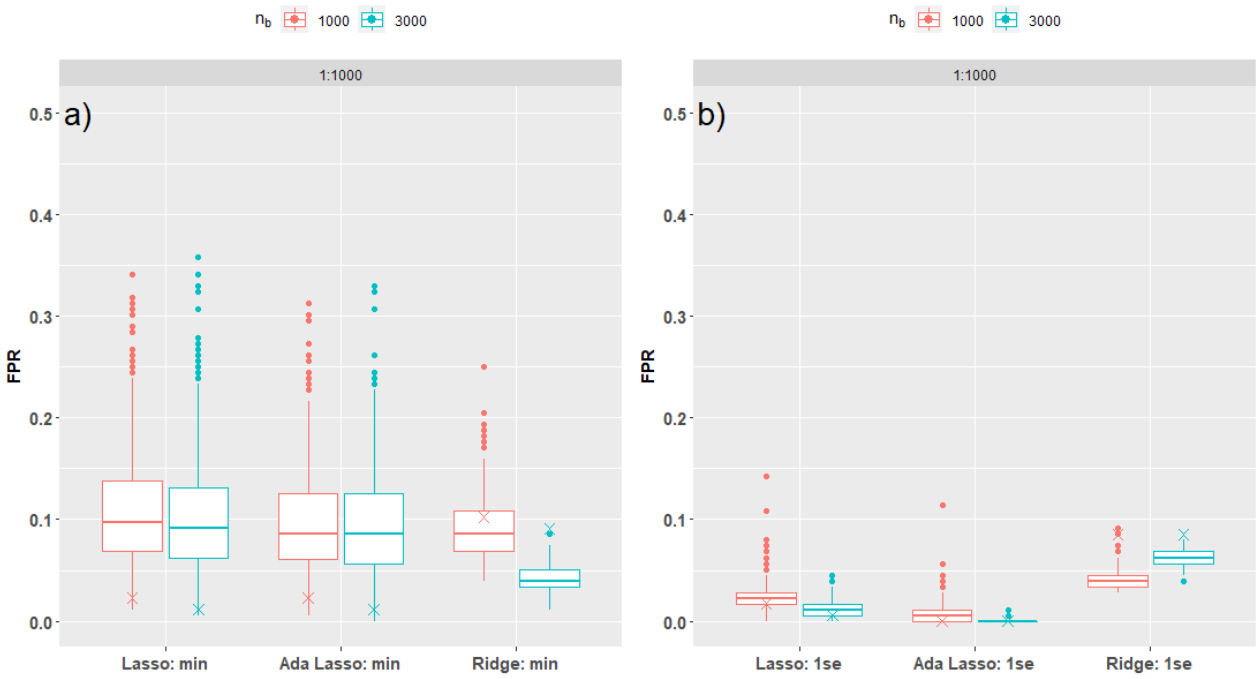


Figure 4.14: MCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits wwith (a) $\lambda =$ "lambda.min" and (b) $\lambda =$ "lambda.1se", for $n_b = 1000, 3000$ and $I_R =$ 1:1000. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

(Figure 4.14).

In Table 4.5, the mean AUC scores for the lasso and ridge regression did not score any lower than 0.900, whereas with the adaptive lasso the mean AUC scores still ranged from approximately 0.820-0.850 due to the low FPR scores. The mRank and cRank AUC scores with ridge regression scored above 0.950 for all sample sizes, however the mRank, cRank and clustering AUC scores with adaptive lasso did not score higher than 0.750, unlike the AUC scores with mPdrop and cPdrop (Table 4.5). In Figure 4.15, ridge regression exhibits substantially less variation under "lambda.1se" than under "lambda.min", and cRank does not score above the median AUC values for all regularised models under "lambda.1se", although the cRank AUC scores for ridge regression are higher than the respective AUCs

under “lambda.min”.

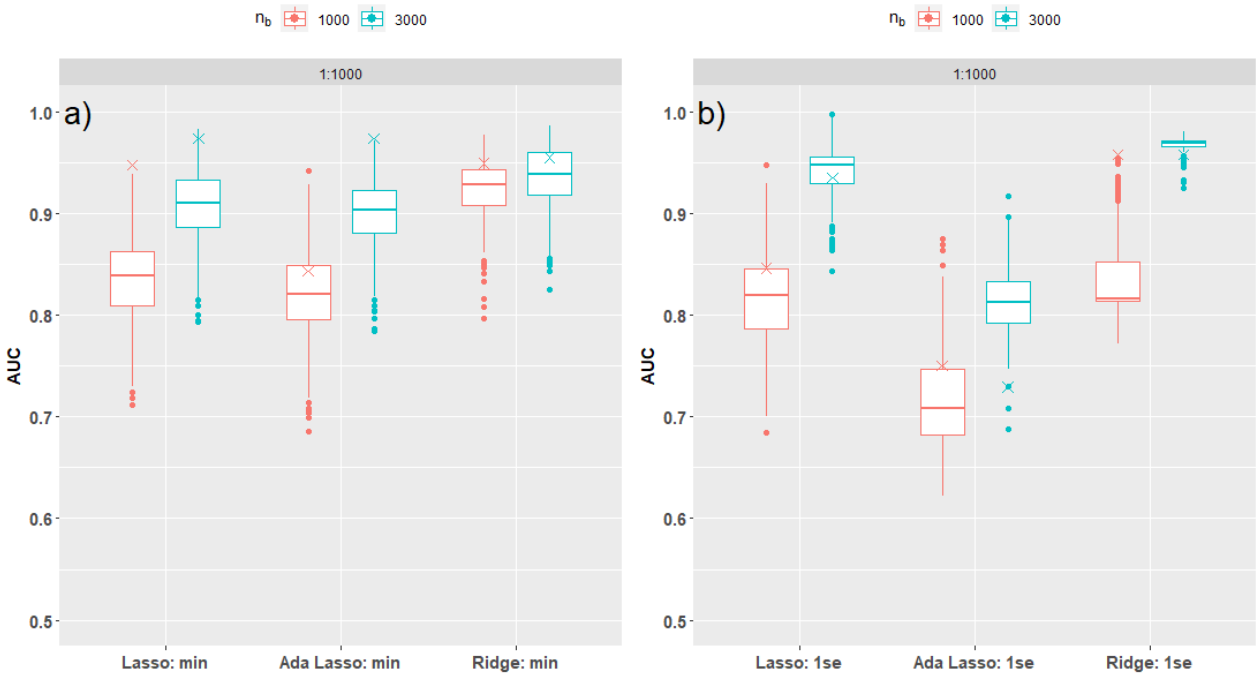


Figure 4.15: MCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits with (a) $\lambda =$ “lambda.min” and (b) $\lambda =$ “lambda.1se”, for $n_b = 1000, 3000$ and $I_R =$ 1:1000. The marker “$\times$” corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Table 4.5: MCOR-24 - Variable selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$ and $\lambda =$ "lambda.1se". The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: 1se	1000	0.817 (0.044)	0.830	0.947	0.83	0.933	0.830
	2000	0.890 (0.031)	0.882	0.882	0.882	0.882	0.882
	3000	0.942 (0.026)	0.973	0.973	0.973	0.971	0.973
	4000	0.932 (0.022)	0.935	0.929	0.929	0.941	0.929
	5000	0.942 (0.021)	0.935	0.935	0.935	0.947	0.977
Ada Lasso: 1se	1000	0.715 (0.048)	0.646	0.843	0.848	0.936	0.771
	2000	0.785 (0.036)	0.864	0.864	0.864	0.884	0.882
	3000	0.814 (0.036)	0.973	0.973	0.973	0.973	0.973
	4000	0.837 (0.031)	0.893	0.935	0.929	0.950	0.929
	5000	0.838 (0.026)	0.935	0.935	0.935	0.935	0.935
Ridge: 1se	1000	0.835 (0.039)	0.956	0.949	-	-	-
	2000	0.962 (0.014)	0.956	0.949	-	-	-
	3000	0.968 (0.006)	0.972	0.955	-	-	-
	4000	0.967 (0.005)	0.949	0.926	-	-	-
	5000	0.966 (0.012)	0.941	0.952	-	-	-

4.4 Discussion

We focus our results when our simulation models are under severe class-imbalance ($I_R = 1{:}1000$) in order to thoroughly assess the performance the regularised OLR models and automatic selection based methods. It is apparent that the automatic selection methods did not recover most signal covariates when faced with a larger set of covariates. Under MUNCOR-24 and MCOR-24 it is evident that the automatic selection methods recovered a lower proportion of signal covariates at sample sizes $n_b = 1000$, 2000 and 3000, relative to MUNCOR-12 and MCOR-12. Under MUNCOR-24 and MCOR-24, the automatic selection

based method TPR scores often placed below the mean/median TPR value for the lasso and adaptive lasso ensembles, whether correlation was present in the simulation models or not. The mean TPR scores for each regularised OLR model under MCOR-12 were very similar to the respective scores under MUNCOR-12. There is a high possibility that under MUNCOR-24 and MCOR-24 the covariates that correspond to low magnitude regression coefficients are frequently zeroed-out across the ensemble of fits resulting in high P_{Drop} values, therefore being screened out by the automatic selection methods during the last step of the algorithm. It is important to note that the noise to signal ratios is 1:2 when $p = 100$ and $p = 200$ covariates.

Under MCOR-12 a higher proportion of noise covariates were selected for all regularised OLR models relative to MUNCOR-12, however each automatic selection method managed to filter a high proportion of noise covariates and the respective FPR scores were much lower than the mean/median FPRs. The mean FPR scores corresponding to the lasso and adaptive lasso under MUNCOR-24 were similar between the respective scores under MCOR-24, although for ridge regression the mean FPR scores are lower under MUNCOR-24 relative to MCOR-24.

According to both the mean AUC scores for regularised OLR models and threshold based AUC scores, the overall balance between TPR and FPR was excellent, with stable variable rankings. Under each simulation model, all regularised OLR models attained a mean AUC score above 0.800 for all sample sizes, except ridge regression under MUNCOR-24. Mean AUC scores generally increased as sample size increases, although there was considerable variation across each individual model. With the implementation of both ensemble and automatic threshold based methods, variable ranking and selection was superior to a single regularised model fit, and this was evident across each simulation model. Automatic threshold based AUC scores were reasonable under all simulation models, most often scoring above 0.800 and usually attaining a score similar or higher than the mean/median AUC

score. Across all simulation models, we did not observe any substantial variable selection improvement between the adaptive lasso relative to the standard lasso, however the adaptive lasso often screened out a higher proportion of noise covariates on average. We also did not observe any automatic threshold based method consistently outperforming another method.

Under MCOR-24 with "lambda.1se", the mean TPR scores corresponding to the lasso and adaptive lasso were lower for all sample sizes relative to the corresponding scores under MCOR-24 with "lambda.min", yet this was not the case for ridge regression except at $n_b = 1000$. The mRank and mPdrop TPR scores corresponding to the lasso failed to recover a substantial amount of signal covariates at $n_b = 1000$ and 2000, however the TPR scores improved with increasing sample size. The clustering method corresponding to the lasso did not recover at more than half of the signal covariates at $n_b = 1000$ and 2000. For the adaptive lasso, each automatic threshold method often failed to recover at more than half of the signal covariates. Clearly ridge regression is superior at recovering signal covariates, especially at "lambda.1se".

The mean FPR scores for the lasso and adaptive lasso and the respective automatic selection method FPR scores were substantially lower on average relative to each simulation model at "lambda.min", although the FPR scores corresponding the ridge regression ensemble and the respective automatic threshold methods were higher.

The overall mean AUC scores were excellent, where the lasso and ridge regression scored above 0.900 for all sample sizes, and 0.800 for the adaptive lasso. The mean AUC scores corresponding to the lasso and ridge regression were higher than the respective scores at MCOR-24 with "lambda.min". We note that at $n_b = 1000$ and 2000, the automatic threshold based AUC scores with the lasso and adaptive lasso are higher at "lambda.min", however the AUC scores are relatively similar to each other with the lasso and ridge regression. The automatic threshold based methods with the adaptive lasso fails to attain reasonable variable selection performances especially mRank, cRank and clustering with "lambda.1se".

Chapter 5

A Case Study on Mesoscale Spatio-Temporal Wildland Fire Occurrence Models in British Columbia

In this chapter we delve into the research conducted by Nadeem et al. (2020) that developed spatio-temporal models for daily human-caused and lightning-caused wildland fire occurrences in British Columbia. We describe the data compilation process along with the description of covariates and the framework used to develop the wildland fire models, along with the application of our automatic threshold based methods on these models.

5.1 Background

Wildland fires occur at random patterns caused by lightning strikes and human negligence on a daily basis during a typical fire season. Wildland fires account for an average

of 2.5 million hectares of burnt area across Canada annually, costing the government of Canada up to $1.5 billion per year (nat, 2021). The province of British Columbia is a large contributor to Canadian wildland fires, as 70% of its land area contains coniferous forest or grasslands that are potentially flammable (Nadeem et al., 2020). The destructive behaviour and occurrences of wildland fires have been studied for decades, where several predictive modelling frameworks have been implemented such as the negative binomial model (Bruce, 1963), and the Poisson model (Cunningham and Martell, 1973). With the development of remote automated fire weather stations and lightning location detectors in the 1980s, this allowed for a higher resolution of fire-weather data and more sophisticated modelling techniques (Nadeem et al., 2020). A discretised approach to model wildland fire occurrences was introduced in the work of Brillinger et al. (2003), and was implemented by Nadeem et al. (2020), where the wildland fire occurrence process is assumed to be a spatio-temporal point process with an inhomogeneous conditional intensity function that depends on a variation of predictors.

5.2 Data Compilation

Nadeem et al. (2020) compiled 34 years of data (1981-2014) collected from weather stations, and other database inventories. The data corresponds to geographic, vegetation, ecumene, surface fire weather, atmospheric stability, time periods, lightning and baseline risk variables. A subset of the variables are listed below:

Geographic: Longitude, Latitude, Elevation, EcoZone;

Vegetation: Vegetated Proportion, Treed Proportion, Proportion of Conifer Cover;

Ecumene: Road Length, Population, Wildland-Urban Interface (WUI) Area;

Lightning: Lightning Strikes (strikes counted in the previous 24-h), Lightning Indicator;

Surface Fire Weather: Temperature, Relative Humidity, Wind Speed, Precipitation, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Fire Weather Index (FWI);

Atmospheric Stability: Showalter Index, 500 mb Anomaly (500 mb (hPa) geopotential height anomaly).

Nadeem et al. (2020) considered three baseline risk covariates, ten geographic, two regarding time periods, fifteen measures of vegetation, eight for ecumene, six for atmospheric stability, five for lightning strikes, and thirty-two measures pertaining to surface fire weather. The spatial domain of British Columbia is represented by 2541 20×20 km spatial units in the National Forest Inventory (NFI) grid. Each unit represents a 24-h time period (starting at midnight) bounded by the wildland fire season beginning from 16 March to 14 October where 99% of wildland fires occurred in British Columbia. The data corresponding to the explanatory variables were binned by day and cell or interpolated to the centroid of each space-time unit (Nadeem et al. 2020).

5.3 Wildland Fire Occurrence Modelling with the Logistic-Lasso

Three wildland fire occurrence models were developed by Nadeem et al. (2020) using the logistic-lasso modelling framework. The three models are described below :

A Predicted Lightning-Caused Fire (PLCF) model to forecast lightning caused fires, in part utilizing weather and atmospheric stability measures which can be computed

from medium-term numerical weather model forecasts;

An Observed Lightning-Caused Fire (OLCF) model in part utilizing recent lightning strike and weather observations to potentially nowcast fires that have occurred (although they may or may not have been reported);

A Human-Caused Fire (HCF) model which can nowcast and forecast human caused fires utilizing observed or forecasted surface weather conditions.

The PLCF and HCF models were trained on data from 1981-2008, leaving the years 2009-2014 for testing. The OLCF was trained on data from years 1999-2008 and 2010-2013, leaving the years 2009 and 2014 for testing. Only 67, 69 and 82 covariates were considered for the OLCF, PLCF and HCF models respectively. A large volume of data was compiled due to the the nature of the spatio-temporal domain, resulting in roughly 18 million observations (2541 voxels $\times$ 34 years $\times$ 214 days) where only 0.23% and 0.18% contain lightning caused and human caused wildfires respectively (Nadeem et al., 2020). The case-control class-imbalance ratio for the HCF, PLCF and OLCF models are 1:469, 1:433 and 1:504 respectively.

The wildfire occurrence in each space-time unit is modelled as a Bernoulli outcome (Y = 1). Nadeem et al. (2020) fit an ensemble of $M = 500$ logistic-lasso models on balanced datasets achieved via response-based sampling. The predictive skill of the HCF, OLCF and PLCF models were assessed through ROC characteristics, provincial-scale time series and temporal and spatial residuals (Nadeem et al., 2020). The predictive skill of the corresponding models excelled in terms of sensitivity (TPR), specificity (1 - FPR) and AUC scores, where all models attained an AUC score above 0.90. The PLCF, OLCF and HCF models respectively attained a sensitivity score of 0.898, 0.905, and 0.866 and their respective specificity scores are 0.814, 0.861, and 0.811 (Nadeem et al., 2020).

5.4 Variable Ranking and Selection via a Heuristic Approach

Nadeem et al. (2020) implemented their variable ranking and selection method to identify important covariates in HCF, PLCF and OLCF models. The subset of important covariates were selected manually based on the visual representation between the relationship of the $Rank(x_i)$ and P_{Drop} scores obtained from their ranking based method. A threshold was selected corresponding to the P_{Drop} values, to separate the ranked covariates into two clusters representing important and non-important covariates. There were 23, 20 and 32 covariates deemed important corresponding to the OLCF, HCF and PLCF models respectively, according to the threshold rule depicted in Figure 5.1. Identifying a set of important covariates via visual inspection can become infeasible as inconsistencies and ambiguity may arise in the selection process. This opens the path to implement our variable ranking and selection algorithm which considers various automatic threshold methods.

5.5 Variable Ranking and Selection via Automatic Thresholding Methods

We would like to examine whether the heuristic approach taken in Nadeem et al. (2020) attained a subset of viable covariates for each wildland fire model. This is done by refitting $M = 500$ logistic-lasso models (with "lambda.1se"), on independent balanced datasets where we permute a subset of covariates. More specifically, we did not permute the top 36, 32 and 27 covariates for HCF, PLCF, and OLCF models respectively from the ranking tables reported in Nadeem et al. (2020). Our resultant rankings with a subset of permuted covariates for the HCF, PLCF and OLCF models are respectively denoted as "HCF permuted", "PLCF

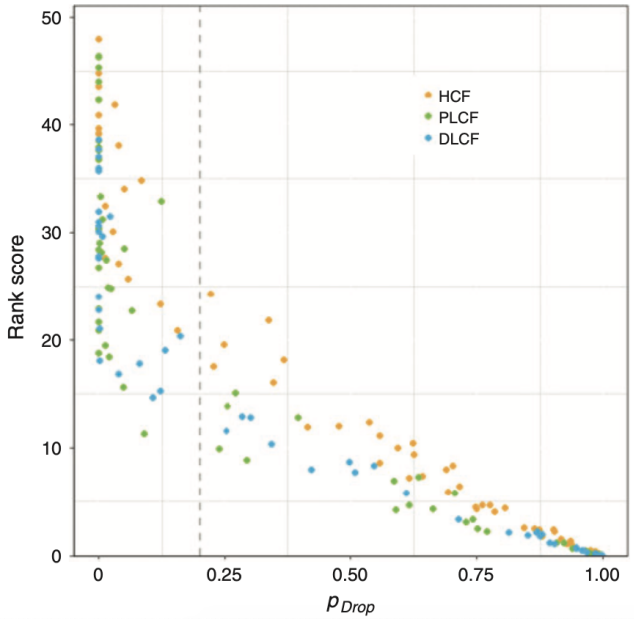


Figure 5.1: (Nadeem et al., 2020) The relationship of $Rank(x_i)$ and P_{Drop}, where each point corresponds to a covariate in each model. The dashed line is determined around 0.2 threshold where 23, 32, and 20 covariates were selected to the left of the threshold for the OLCF, PLCF and HCF models respectively.

permuted" and "OLCF permuted". The ranking tables reported in Nadeem et al. (2020) corresponding to the HCF, PLCF and OLCF models are respectively denoted as "HCF all unpermuted", "PLCF all unpermuted", and "OLCF all unpermuted". Permuting a covariate destroys its relationship with the response vector, and is also implemented to assess variable importance for classification tasks with Random Forests (Breiman, 2001). If the automatic threshold methods retain the subset of covariates that were not permuted, this would provide supporting evidence that these covariates have a large likelihood of being the true important covariate. Likewise if a covariate is permuted and is not selected via the algorithm, then this confirms that this covariate has a large likelihood that it is truly not important.

We implemented the automatic threshold methods to the full model rankings without any permuted covariates and compared the subset of covariates we attained with the subset of "important" covariates determined with the threshold $P_{Drop} = 0.2$ in Nadeem et al.

(2020). Likewise we applied the automatic threshold methods to the full model rankings with permuted covariates and compared our results attained without permuted covariates.

In Table 5.1, each automatic threshold method often recovered a similar amount of covariates relative to the threshold $P_{Drop} = 0.2$. Only mRank recovered the same amount of covariates as with the threshold of $P_{Drop} = 0.2$, whereas mRank and cPdrop selected the most covariates (26). A smaller subset of covariates were attained by all thresholds for the HCF rankings with a subset of permuted covariates. The unpermuted covariates that rank above position 20 have a much higher likelihood of being noise.

Table 5.1: Number of selected covariates by each automatic threshold for the HCF, PLCF and OLCF models, generated from our ranking results with permuted covariates along with the results reported in Nadeem et al. (2020).

Rank Table	$P_{Drop} = 0.2$	mRank	mPdrop	cRank	cPdrop	clustering
HCF all unpermuted	20	26	20	22	26	21
HCF permuted	-	11	15	16	20	12
PLCF all unpermuted	32	6	37	30	37	37
PLCF permuted	-	28	32	28	28	28
OLCF all unpermuted	23	5	23	21	27	13
OLCF permuted	-	25	24	21	24	25

For the PLCF model, cRank selected 30 covariates before permutation, only two less than the threshold $P_{Drop} = 0.2$, whereas mPdrop, cPdrop and clustering selected the most covariates (37) and mRank selected only 6 covariates. The methods mRank, cRank, cPdrop, and clustering recovered 28 covariates after permutation, whereas mPdrop recovered 32, the same amount selected by $P_{Drop} = 0.2$ when covariates were not permuted. The unpermuted covariates ranked above position 32 have a very high likelihood of being noise.

With regards to the OLCF model, prior to covariate permutation, only cPdrop selected more covariates (27) than the threshold at $P_{Drop} = 0.2$, whereas mPdrop selected the same amount of covariates as the threshold (23). We see that mRank selected 5 covariates, a

similar performance when implemented with the PLCF model. After covariate permutation, mRank and clustering selected the highest amount of covariates (25). Overall, each automatic threshold method recovered a similar amount of covariates. Nonpermuted covariates ranked above position 25 have a high likelihood of being noise.

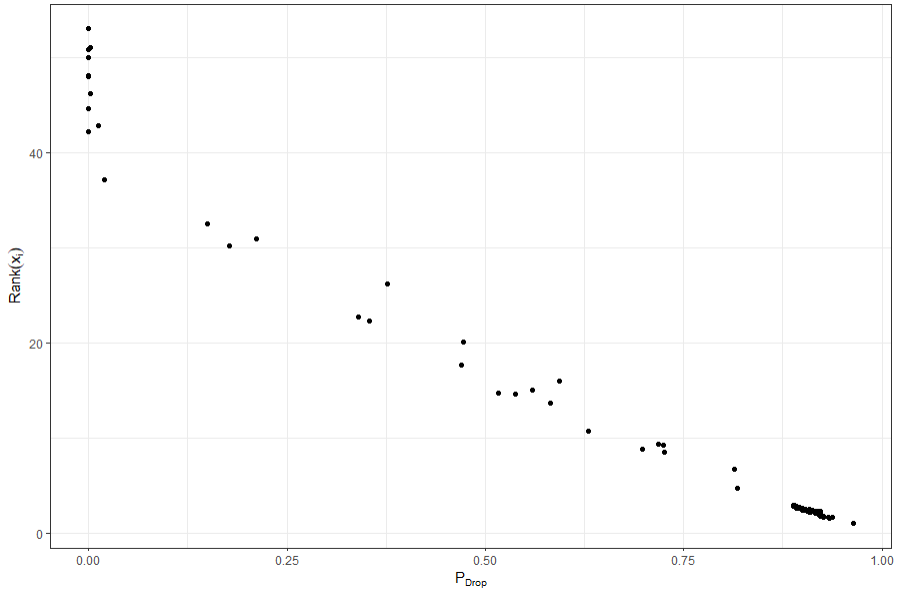


Figure 5.2: $Rank(x_i)$ vs P_{Drop} scores for the HCF model with a subset of permuted covariates

Overall, we observed that the automatic threshold methods had not selected any permuted (noise) covariates. These observations fall in line with our simulation study results where these threshold based methods frequently filtered most noise covariates present in the regularised OLR model, especially with high correlation present between a subset of noise and signal covariates. As for the recovery of signal covariates, we have also seen in our simulation study that a high proportion of signal covariates were recovered for each simulation model especially for large a enough sample size, which is the case for the wildland fire data. The balanced sample size for the HCF and PLCF models are 45,050 and 20,256 for the OLCF model. The threshold based methods generally agreed with each other after covariate permutation. Likewise this was the case in our simulation study, so we can be assured that

the majority of signal covariates were recovered and noise were filtered out regarding these wildland fire models.

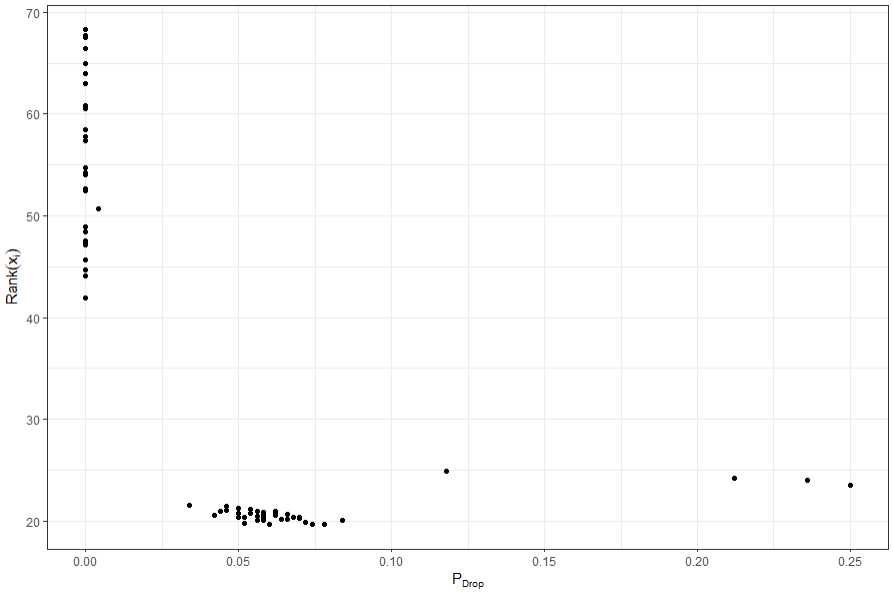


Figure 5.3: $Rank(x_i)$ vs P_{Drop} scores for the PLCF model with a subset of permuted covariates

From a practical point of view, since all permuted covariates were filtered out, it may be more optimal to "choose" the set of covariates via the automatic threshold method that recovered the most covariates, since the likelihood of selecting a false positive should be lower than the likelihood of a false negative (screening out a signal covariate). For example, under the HCF model mRank selected 11 covariates whereas cPdrop selected 20, this could potentially be 9 truly important covariates not selected. The difference in TPR would potentially be much lower if 9 covariates were selected than 20, whereas the difference in FPR if 9 covariates were selected over 20 would be substantially smaller. This is assuming the top 37 ranked covariates are truly important.

In figures 5.2; 5.3; 5.4, it is evident that there is an elbow that separates the covariates into two clusters, simplifying the ability of selecting a set of important covariates based upon the visual representation of the $Rank(x_i)$ and P_{Drop} scores with permuted covariates. Here,

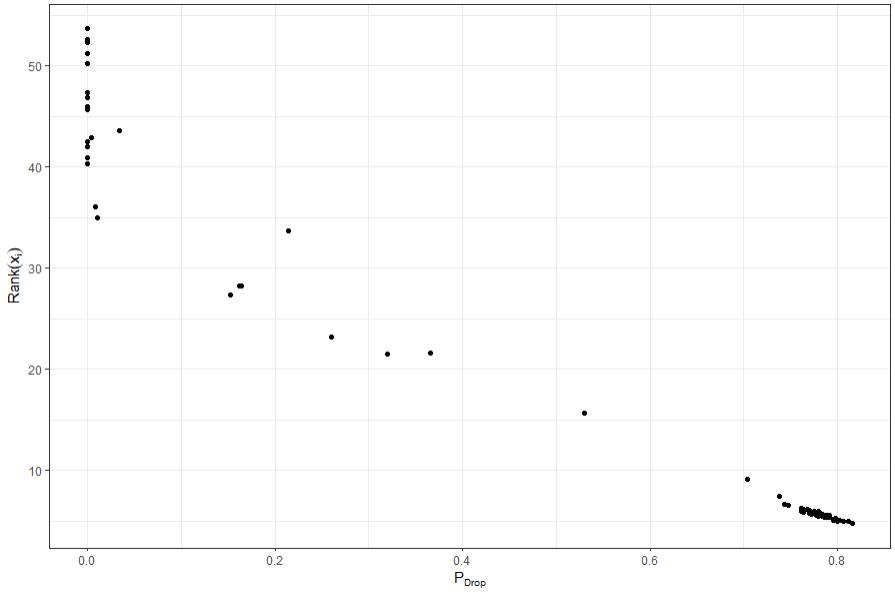


Figure 5.4: $Rank(x_i)$ vs P_{Drop} scores for the OLCF model with a subset of permuted covariates

we are able to compare the number of covariates recovered by each automatic threshold method and threshold selected by a practitioner. For example, in Figure 5.3 we can select all covariates above the threshold at $Rank(x_i) = 40$, which recovered 28 covariates. Similarly mRank, cRank, cPdrop and clustering recovered the same amount.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis demonstrated in several respects the advantages and contributions our automatic variable ranking and selection algorithm exemplified, via the results of our simulation study. The key contributions include but are not limited to; i) data reduction of a big volume of data and re-balancing class distributions using response-based sampling without the loss of information; ii) compatibility of our algorithm with both l_1 and l_2 OLR regularisation models; iii) generating stable variable rankings under the presence of high correlation between a subset of signal and noise covariates using an ensemble of regularised OLR models; iv) the implementation of automatic selection methods are superior at recovering and screening out a high proportion of signal and noise covariates under the proposed conditions, relative to an individual OLR regularised model fit.

Our thesis also illustrated that neither the implementation of the lasso, adaptive lasso or ridge regression consistently outperformed one another in terms of stable variable rankings (via TPR, FPR and AUC scores), likewise for each automatic threshold method. We did not find any sensible reason to opt for the (two-stage) adaptive lasso over the standard lasso

as it only marginally improved upon the lasso, and would also be more computationally expensive. The results obtained with ridge regression are evidence that automatic variable selection can be applied with l_2 regularisation methods, and also excels with and without correlation present. Our results show that variable ranking and selection mean AUC increased for ridge regression at "lambda.1se". Likewise for the lasso, however threshold based methods were not optimal for smaller sample sizes. Further studies can be conducted to investigate other potential OLR regularisation methods at varying λ values, along with other automatic selection methods. Overall, we recommend the implementation of ridge regression at "lambda.1se" in our algorithm by practitioners given an OLR model with a large set of correlated covariates (say 200) and the standard lasso with a smaller set of covariates (say 100).

6.2 Future Work

In this thesis, we thoroughly investigated variable ranking and selection via specific OLR regularisation methods such as the lasso, adaptive lasso and ridge regression along with various automatic threshold based variable selection methods. However, the field of variable ranking and selection is vast, and other regularised modelling and automatic threshold based variable selection methods should be studied to make further recommendations and conclusions.

This thesis provides evidence that both the lasso and adaptive lasso typically select a substantial amount of noise covariates regardless if high correlation between a subset of signal and noise covariates is present in the model. Although the automatic threshold methods frequently filter out the majority of the noise covariates present in the models, it is beneficial to recognize whether other regularisation methods would obtain lower FPR scores on average, whilst retaining the true set of covariates better than what we had studied in

this thesis. The elastic net and the precision lasso (Wang et al., 2019) are other methods to potentially consider. The precision lasso utilizes the covariance and inverse covariance matrices of the covariates to perform regularisation (automatic variable selection). The precision lasso has been shown to outperform the lasso, elastic net and Minimax Concave Penalty (MCP) regression under the presence of correlated and linearly dependent covariates (Wang et al., 2019). The broken adaptive ridge regression is another method that may be studied as well.

Our results also showed that hyper-parameters such as λ had a large impact on the variable ranking and selection performance for each OLR regularisation method. Effect of λ can be rigorously examined by further considering values 0.25, 0.50 and 0.75 standard deviations away from "lambda.min". The hyper-parameters such as γ and α appearing the adaptive lasso and elastic net respectively are also in contention to be examined.

In our simulation study, we considered $M = 500$ regularised OLR model fits to attain stable variable rankings. However, this may become computationally expensive especially when considering different values of hyper-parameters or several regularisation modelling methods in a simulation study. We find evidence (not reported here) that only a fraction of model fits ($M = 100$) are sufficient to generate stable variable rankings. Also, one may attempt to determine the value of M such that stable variable rankings are achieved instead of a very large number of models thereby only marginally increasing selection accuracy at the expense of a higher computational cost.

Despite the threshold based variable selection methods preforming relatively well, there may be other methods that could be an improvement on these methods we studied, such as segmented (broken-stick) regression or adaptive splines. These methods can also be implemented to find a threshold (change point) using the relationship of $Rank(x_i)$ and P_{Drop} as we have done with complete-linkage clustering.

The focus of our thesis was variable ranking and selection. However, assessing the true

ordering of the covariate rank positions attained by our algorithm via a simulation study may be of interest in applied work. This can achieved by assessing the Somers' D score (Somers, 1962). A reasonable Somer's D score in a simulation study would indicate that the rank positions of the covariates attained by our algorithm tend to recover true ordering of relative importance of the covariates. Acquiring this knowledge can potentially aid practitioners with regards to study design, data collection and further research and development.

Lastly, it will be interesting to explore our methodology in cases where $p >> n$, which is referred to as the "curse of dimensionality" leaving the data matrix $\mathbf{x}$ ill-posed. This phenomenon frequently appears in genomics datasets, where regularized regression is a popular tool to circumvent this issue by finding a sparse solution using, for instance, the lasso penalty (Algamal, 2017; Shi et al., 2012)

Appendix

Table A1: MUNCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.966 (0.052)	1.000	1.000	1.000	1.000	1.000
	2000	1.000 (0.006)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.962 (0.054)	1.000	1.000	1.000	1.000	1.000
	2000	0.999 (0.007)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ridge: min	1000	0.860 (0.090)	1.000	1.000	-	-	-
	2000	0.964 (0.052)	1.000	1.000	-	-	-
	3000	0.987 (0.032)	1.000	1.000	-	-	-
	4000	0.964 (0.050)	1.000	1.000	-	-	-
	5000	1.000 (0.000)	1.000	1.000	-	-	-

Table A2: MUNCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.986 (0.038)	1.000	1.000	1.000	1.000	1.000
	2000	0.998 (0.014)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.985 (0.040)	1.000	1.000	1.000	1.000	1.000
	2000	0.998 (0.014)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	0.250
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ridge: min	1000	0.913 (0.081)	1.000	1.000	-	-	-
	2000	0.905 (0.076)	1.000	1.000	-	-	-
	3000	0.978 (0.039)	1.000	1.000	-	-	-
	4000	0.974 (0.045)	1.000	1.000	-	-	-
	5000	0.976 (0.039)	1.000	1.000	-	-	-

Table A3: MUNCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.933 (0.042)	0.917	0.917	0.917	0.917	0.917
	2000	0.998 (0.014)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.932 (0.042)	0.917	0.917	0.917	0.917	0.917
	2000	0.998 (0.014)	1.000	1.000	1.000	1.000	1.000
	3000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ridge: min	1000	0.843 (0.089)	0.917	1.000	-	-	-
	2000	0.969 (0.056)	0.917	1.000	-	-	-
	3000	0.970 (0.043)	1.000	1.000	-	-	-
	4000	0.991 (0.029)	1.000	1.000	-	-	-
	5000	0.985 (0.033)	1.000	1.000	-	-	-

Table A4: MUNCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.173 (0.130)	0.011	0.011	0.011	0.011	0.011
	2000	0.100 (0.081)	0.000	0.000	0.000	0.011	0.057
	3000	0.141 (0.094)	0.000	0.011	0.000	0.034	0.034
	4000	0.104 (0.082)	0.000	0.000	0.000	0.023	0.000
	5000	0.068 (0.073)	0.000	0.000	0.000	0.000	0.045
Ada Lasso: min	1000	0.159 (0.109)	0.011	0.011	0.011	0.011	0.011
	2000	0.093 (0.075)	0.000	0.000	0.000	0.011	0.057
	3000	0.131 (0.088)	0.000	0.011	0.000	0.034	0.034
	4000	0.094 (0.078)	0.000	0.000	0.000	0.011	0.000
	5000	0.059 (0.072)	0.000	0.000	0.000	0.000	0.045
Ridge: min	1000	0.013 (0.017)	0.011	0.034	-	-	-
	2000	0.000 (0.001)	0.000	0.045	-	-	-
	3000	0.000 (0.001)	0.011	0.114	-	-	-
	4000	0.000 (0.000)	0.000	0.080	-	-	-
	5000	0.000 (0.000)	0.000	0.091	-	-	-

Table A5: MUNCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.159 (0.125)	0.000	0.011	0.000	0.034	0.057
	2000	0.191 (0.126)	0.000	0.034	0.000	0.034	0.034
	3000	0.062 (0.061)	0.000	0.000	0.000	0.000	0.000
	4000	0.146 (0.110)	0.000	0.011	0.000	0.080	0.000
	5000	0.072 (0.070)	0.000	0.000	0.000	0.000	0.057
Ada Lasso: min	1000	0.144 (0.104)	0.000	0.011	0.000	0.034	0.057
	2000	0.177 (0.110)	0.000	0.011	0.000	0.034	0.034
	3000	0.054 (0.057)	0.000	0.000	0.000	0.000	0.000
	4000	0.134 (0.099)	0.000	0.011	0.000	0.023	0.000
	5000	0.063 (0.069)	0.000	0.000	0.000	0.000	0.057
Ridge: min	1000	0.010 (0.013)	0.011	0.148	-	-	-
	2000	0.001 (0.003)	0.000	0.114	-	-	-
	3000	0.000 (0.000)	0.000	0.000	-	-	-
	4000	0.000 (0.001)	0.000	0.091	-	-	-
	5000	0.000 (0.000)	0.000	0.057	-	-	-

Table A6: MUNCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.138 (0.101)	0.000	0.000	0.000	0.023	0.000
	2000	0.191 (0.126)	0.000	0.034	0.000	0.034	0.034
	3000	0.087 (0.072)	0.000	0.000	0.000	0.000	0.057
	4000	0.078 (0.068)	0.000	0.000	0.000	0.000	0.000
	5000	0.067 (0.060)	0.000	0.011	0.011	0.011	0.011
Ada Lasso: min	1000	0.126 (0.085)	0.000	0.000	0.000	0.023	0.057
	2000	0.177 (0.110)	0.000	0.011	0.000	0.034	0.034
	3000	0.080 (0.069)	0.000	0.000	0.000	0.000	0.000
	4000	0.069 (0.068)	0.000	0.000	0.000	0.000	0.000
	5000	0.057 (0.057)	0.000	0.000	0.000	0.011	0.011
Ridge: min	1000	0.008 (0.013)	0.000	0.091	-	-	-
	2000	0.001 (0.003)	0.000	0.091	-	-	-
	3000	0.000 (0.000)	0.000	0.057	-	-	-
	4000	0.000 (0.000)	0.000	0.023	-	-	-
	5000	0.000 (0.000)	0.011	0.045	-	-	-

Table A7: MUNCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.679 (0.118)	0.333	0.708	0.625	0.708	0.625
	2000	0.750 (0.084)	0.542	0.667	0.667	0.708	0.667
	3000	0.813 (0.059)	0.708	0.792	0.708	0.875	0.708
	4000	0.835 (0.05)	0.792	0.792	0.792	0.792	0.833
	5000	0.856 (0.032)	0.833	0.833	0.833	0.833	0.833
Ada Lasso: min	1000	0.675 (0.118)	0.333	0.708	0.625	0.708	0.625
	2000	0.746 (0.084)	0.542	0.667	0.667	0.708	0.667
	3000	0.810 (0.059)	0.708	0.792	0.708	0.833	0.875
	4000	0.833 (0.050)	0.792	0.792	0.792	0.792	0.833
	5000	0.855 (0.031)	0.833	0.833	0.833	0.833	0.833
Ridge: min	1000	0.356 (0.064)	0.500	0.833	-	-	-
	2000	0.378 (0.072)	0.875	0.917	-	-	-
	3000	0.358 (0.102)	0.667	0.875	-	-	-
	4000	0.403 (0.057)	0.792	0.875	-	-	-
	5000	0.503 (0.060)	0.833	0.833	-	-	-

Table A8: MUNCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.63 (0.101)	0.458	0.542	0.458	0.583	0.458
	2000	0.815 (0.049)	0.792	0.792	0.792	0.792	0.875
	3000	0.787 (0.050)	0.750	0.750	0.750	0.750	0.750
	4000	0.831 (0.038)	0.792	0.833	0.792	0.833	0.792
	5000	0.903 (0.033)	0.875	0.917	0.875	0.917	0.917
Ada Lasso: min	1000	0.626 (0.099)	0.458	0.542	0.458	0.542	0.458
	2000	0.813 (0.049)	0.792	0.792	0.792	0.792	0.875
	3000	0.785 (0.050)	0.750	0.750	0.750	0.750	0.750
	4000	0.829 (0.038)	0.750	0.833	0.792	0.833	0.833
	5000	0.901 (0.033)	0.875	0.917	0.875	0.917	0.917
Ridge: min	1000	0.380 (0.074)	0.458	0.792	-	-	-
	2000	0.355 (0.077)	0.792	0.875	-	-	-
	3000	0.488 (0.067)	0.750	0.875	-	-	-
	4000	0.458 (0.058)	0.833	0.875	-	-	-
	5000	0.390 (0.051)	0.917	0.958	-	-	-

Table A9: MUNCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.615 (0.100)	0.458	0.458	0.458	0.583	0.458
	2000	0.741 (0.074)	0.667	0.667	0.667	0.708	0.708
	3000	0.813 (0.069)	0.667	0.750	0.750	0.833	0.667
	4000	0.846 (0.058)	0.833	0.833	0.833	0.833	0.833
	5000	0.860 (0.042)	0.833	0.833	0.833	0.833	0.833
Ada Lasso: min	1000	0.610 (0.098)	0.458	0.458	0.458	0.583	0.458
	2000	0.736 (0.074)	0.667	0.667	0.667	0.708	0.708
	3000	0.810 (0.068)	0.667	0.750	0.667	0.833	0.750
	4000	0.843 (0.058)	0.750	0.833	0.833	0.833	0.833
	5000	0.857 (0.042)	0.833	0.833	0.833	0.833	0.833
Ridge: min	1000	0.361 (0.095)	0.542	0.958	-	-	-
	2000	0.348 (0.081)	0.708	0.875	-	-	-
	3000	0.434 (0.067)	0.792	0.958	-	-	-
	4000	0.407 (0.095)	0.833	0.833	-	-	-
	5000	0.399 (0.068)	0.833	0.917	-	-	-

Table A10: MUNCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.104 (0.077)	0.000	0.011	0.006	0.011	0.006
	2000	0.088 (0.055)	0.000	0.006	0.006	0.006	0.006
	3000	0.118 (0.060)	0.006	0.017	0.006	0.028	0.006
	4000	0.104 (0.057)	0.000	0.006	0.000	0.011	0.040
	5000	0.111 (0.056)	0.000	0.011	0.000	0.028	0.028
Ada Lasso: min	1000	0.099 (0.073)	0.000	0.006	0.006	0.011	0.006
	2000	0.085 (0.053)	0.000	0.006	0.006	0.006	0.006
	3000	0.112 (0.056)	0.006	0.017	0.006	0.028	0.028
	4000	0.099 (0.054)	0.000	0.000	0.000	0.011	0.040
	5000	0.105 (0.051)	0.000	0.006	0.000	0.028	0.006
Ridge: min	1000	0.001 (0.002)	0.000	0.142	-	-	-
	2000	0.000 (0.000)	0.028	0.074	-	-	-
	3000	0.000 (0.000)	0.000	0.097	-	-	-
	4000	0.000 (0.000)	0.000	0.074	-	-	-
	5000	0.000 (0.000)	0.000	0.062	-	-	-

Table A11: MUNCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.082 (0.070)	0.000	0.000	0.000	0.006	0.000
	2000	0.092 (0.051)	0.000	0.000	0.000	0.006	0.040
	3000	0.124 (0.067)	0.000	0.000	0.000	0.034	0.000
	4000	0.112 (0.062)	0.000	0.006	0.000	0.017	0.000
	5000	0.116 (0.053)	0.000	0.011	0.000	0.023	0.023
Ada Lasso: min	1000	0.080 (0.067)	0.000	0.000	0.000	0.000	0.000
	2000	0.088 (0.048)	0.000	0.000	0.000	0.006	0.040
	3000	0.119 (0.064)	0.000	0.000	0.000	0.023	0.000
	4000	0.106 (0.058)	0.000	0.006	0.000	0.017	0.017
	5000	0.109 (0.049)	0.000	0.011	0.000	0.023	0.023
Ridge: min	1000	0.001 (0.003)	0.000	0.080	-	-	-
	2000	0.000 (0.000)	0.000	0.040	-	-	-
	3000	0.000 (0.000)	0.000	0.119	-	-	-
	4000	0.000 (0.000)	0.000	0.080	-	-	-
	5000	0.000 (0.000)	0.006	0.131	-	-	-

Table A12: MUNCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.088 (0.06)	0.000	0.000	0.000	0.011	0.000
	2000	0.081 (0.053)	0.000	0.006	0.000	0.011	0.011
	3000	0.122 (0.072)	0.000	0.006	0.006	0.006	0.000
	4000	0.116 (0.069)	0.017	0.006	0.017	0.017	0.017
	5000	0.113 (0.057)	0.000	0.000	0.000	0.006	0.000
Ada Lasso: min	1000	0.085 (0.058)	0.000	0.000	0.000	0.011	0.000
	2000	0.077 (0.050)	0.000	0.006	0.000	0.011	0.011
	3000	0.115 (0.068)	0.000	0.006	0.000	0.006	0.006
	4000	0.110 (0.064)	0.000	0.006	0.017	0.017	0.017
	5000	0.106 (0.054)	0.000	0.000	0.000	0.006	0.000
Ridge: min	1000	0.001 (0.003)	0.000	0.142	-	-	-
	2000	0.000 (0.000)	0.000	0.074	-	-	-
	3000	0.000 (0.000)	0.000	0.062	-	-	-
	4000	0.000 (0.000)	0.006	0.028	-	-	-
	5000	0.000 (0.000)	0.000	0.074	-	-	-

Table A13: MCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.852 (0.085)	0.750	0.750	0.75	0.833	0.75
	2000	0.975 (0.044)	0.917	1.000	0.917	1.000	1.000
	3000	1.000 (0.005)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.005)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.836 (0.088)	0.750	0.750	0.75	0.833	0.75
	2000	0.964 (0.05)	0.917	0.917	0.917	1.000	1.000
	3000	0.999 (0.01)	1.000	1.000	1.000	1.000	1.000
	4000	0.999 (0.01)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.004)	1.000	1.000	1.000	1.000	11.000
Ridge: min	1000	0.754 (0.096)	0.833	0.833	-	-	-
	2000	0.858 (0.063)	1.000	1.000	-	-	-
	3000	0.913 (0.066)	1.000	1.000	-	-	-
	4000	0.940 (0.052)	1.000	1.000	-	-	-
	5000	0.965 (0.042)	1.000	1.000	-	-	-

Table A14: MCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.935 (0.078)	0.917	1.000	0.917	1.000	0.917
	2000	0.991 (0.027)	1.000	1.000	1.000	1.000	1.000
	3000	0.977 (0.040)	0.917	1.000	0.917	1.000	0.917
	4000	0.999 (0.011)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.922 (0.088)	0.917	1.000	0.917	1.000	0.917
	2000	0.986 (0.033)	1.000	1.000	1.000	1.000	1.000
	3000	0.971 (0.045)	1.000	1.000	0.917	1.000	1.000
	4000	0.997 (0.015)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ridge: min	1000	0.736 (0.08)	0.833	1.000	-	-	-
	2000	0.891 (0.049)	0.917	1.000	-	-	-
	3000	0.848 (0.067)	0.917	1.000	-	-	-
	4000	0.957 (0.059)	1.000	1.000	-	-	-
	5000	0.992 (0.026)	1.000	1.000	-	-	-

Table A15: MCOR-12 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.911 (0.093)	1.000	1.000	1.000	1.000	1.000
	2000	0.920 (0.071)	0.833	0.917	0.833	0.917	0.833
	3000	0.998 (0.013)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ada Lasso: min	1000	0.894 (0.105)	1.000	1.000	1.000	1.000	0.667
	2000	0.912 (0.078)	0.833	0.917	0.833	0.917	0.833
	3000	0.996 (0.018)	1.000	1.000	1.000	1.000	1.000
	4000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
	5000	1.000 (0.000)	1.000	1.000	1.000	1.000	1.000
Ridge: min	1000	0.757 (0.057)	0.833	1.000	-	-	-
	2000	0.827 (0.034)	0.833	0.917	-	-	-
	3000	0.914 (0.073)	1.000	1.000	-	-	-
	4000	0.963 (0.057)	1.000	1.000	-	-	-
	5000	0.997 (0.016)	1.000	1.000	-	-	-

Table A16: MCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.147 (0.126)	0.000	0.000	0.000	0.045	0.000
	2000	0.215 (0.099)	0.000	0.045	0.000	0.091	0.045
	3000	0.208 (0.099)	0.000	0.000	0.000	0.102	0.000
	4000	0.172 (0.079)	0.000	0.000	0.000	0.023	0.057
	5000	0.207 (0.088)	0.011	0.023	0.011	0.057	0.057
Ada Lasso: min	1000	0.128 (0.105)	0.000	0.000	0.000	0.045	0.000
	2000	0.177 (0.085)	0.000	0.023	0.000	0.045	0.045
	3000	0.171 (0.082)	0.000	0.000	0.000	0.068	0.000
	4000	0.138 (0.073)	0.000	0.000	0.000	0.011	0.057
	5000	0.168 (0.081)	0.011	0.023	0.011	0.057	0.057
Ridge: min	1000	0.022 (0.016)	0.045	0.114	-	-	-
	2000	0.012 (0.014)	0.114	0.114	-	-	-
	3000	0.000 (0.002)	0.023	0.227	-	-	-
	4000	0.002 (0.004)	0.011	0.136	-	-	-
	5000	0.000 (0.002)	0.091	0.091	-	-	-

Table A17: MCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.178 (0.113)	0.000	0.034	0.011	0.068	0.000
	2000	0.216 (0.097)	0.011	0.023	0.023	0.091	0.080
	3000	0.223 (0.110)	0.000	0.045	0.000	0.091	0.000
	4000	0.209 (0.095)	0.000	0.045	0.000	0.125	0.000
	5000	0.179 (0.089)	0.000	0.000	0.000	0.068	0.000
Ada Lasso: min	1000	0.154 (0.095)	0.000	0.011	0.000	0.045	0.000
	2000	0.181 (0.083)	0.011	0.011	0.011	0.068	0.011
	3000	0.186 (0.092)	0.000	0.034	0.000	0.08	0.000
	4000	0.177 (0.082)	0.000	0.045	0.000	0.102	0.000
	5000	0.150 (0.078)	0.000	0.000	0.000	0.011	0.000
Ridge: min	1000	0.021 (0.015)	0.034	0.114	-	-	-
	2000	0.017 (0.013)	0.034	0.148	-	-	-
	3000	0.002 (0.005)	0.034	0.125	-	-	-
	4000	0.000 (0.001)	0.000	0.193	-	-	-
	5000	0.004 (0.006)	0.023	0.091	-	-	-

Table A18: MCOR-12 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.146 (0.127)	0.011	0.011	0.011	0.011	0.011
	2000	0.184 (0.128)	0.000	0.034	0.000	0.102	0.000
	3000	0.148 (0.087)	0.000	0.000	0.000	0.011	0.000
	4000	0.243 (0.110)	0.000	0.023	0.000	0.045	0.011
	5000	0.174 (0.080)	0.000	0.023	0.000	0.045	0.023
Ada Lasso: min	1000	0.122 (0.103)	0.011	0.000	0.011	0.011	0.000
	2000	0.160 (0.106)	0.000	0.023	0.000	0.068	0.000
	3000	0.122 (0.076)	0.000	0.000	0.000	0.000	0.000
	4000	0.205 (0.091)	0.000	0.000	0.000	0.045	0.000
	5000	0.142 (0.070)	0.000	0.023	0.000	0.034	0.057
Ridge: min	1000	0.021 (0.013)	0.034	0.057	-	-	-
	2000	0.006 (0.009)	0.023	0.148	-	-	-
	3000	0.003 (0.006)	0.011	0.102	-	-	-
	4000	0.001 (0.003)	0.011	0.080	-	-	-
	5000	0.002 (0.004)	0.045	0.057	-	-	-

Table A19: MCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.781 (0.059)	0.792	0.792	0.792	0.875	0.875
	2000	0.867 (0.046)	0.875	0.875	0.875	0.875	0.917
	3000	0.898 (0.042)	0.958	0.958	0.708	0.958	0.958
	4000	0.869 (0.040)	0.792	0.833	0.792	0.875	0.792
	5000	0.921 (0.042)	0.833	0.875	0.833	0.958	0.958
Ada Lasso: min	1000	0.747 (0.062)	0.750	0.792	0.75	0.792	0.750
	2000	0.839 (0.048)	0.750	0.875	0.875	0.875	0.625
	3000	0.870 (0.048)	0.708	0.833	0.708	0.958	0.708
	4000	0.852 (0.042)	0.792	0.792	0.792	0.875	0.792
	5000	0.898 (0.046)	0.833	0.833	0.833	0.917	0.958
Ridge: min	1000	0.948 (0.054)	1.000	1.000	-	-	-
	2000	0.868 (0.078)	1.000	1.000	-	-	-
	3000	0.785 (0.098)	0.958	1.000	-	-	-
	4000	0.745 (0.063)	0.917	0.958	-	-	-
	5000	0.691 (0.082)	0.833	1.000	-	-	-

Table A20: MCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.806 (0.064)	0.500	0.792	0.500	0.875	0.917
	2000	0.846 (0.055)	0.792	0.792	0.792	0.833	0.875
	3000	0.840 (0.041)	0.792	0.792	0.792	0.792	0.875
	4000	0.868 (0.043)	0.833	0.833	0.833	0.833	0.833
	5000	0.901 (0.046)	0.750	0.917	0.917	0.958	0.917
Ada Lasso: min	1000	0.765 (0.068)	0.667	0.708	0.708	0.792	0.500
	2000	0.819 (0.054)	0.792	0.792	0.792	0.792	0.875
	3000	0.822 (0.041)	0.792	0.792	0.792	0.792	0.792
	4000	0.851 (0.045)	0.792	0.792	0.833	0.833	0.833
	5000	0.870 (0.053)	0.750	0.792	0.75	0.917	0.917
Ridge: min	1000	0.949 (0.053)	1.000	1.000	-	-	-
	2000	0.833 (0.087)	1.000	1.000	-	-	-
	3000	0.815 (0.075)	0.958	1.000	-	-	-
	4000	0.788 (0.056)	1.000	1.000	-	-	-
	5000	0.716 (0.052)	0.917	1.000	-	-	-

Table A21: MCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.779 (0.069)	0.667	0.917	0.667	0.917	0.667
	2000	0.841 (0.050)	0.792	0.792	0.792	0.792	0.792
	3000	0.920 (0.041)	0.958	0.958	0.958	0.958	0.958
	4000	0.900 (0.041)	0.875	0.875	0.875	0.917	0.875
	5000	0.919 (0.042)	0.875	0.875	0.875	0.917	1.000
Ada Lasso: min	1000	0.738 (0.077)	0.292	0.708	0.708	0.917	0.542
	2000	0.807 (0.052)	0.750	0.750	0.75	0.792	0.792
	3000	0.897 (0.046)	0.958	0.958	0.958	0.958	0.958
	4000	0.877 (0.043)	0.792	0.875	0.875	0.917	0.875
	5000	0.891 (0.042)	0.875	0.875	0.875	0.875	0.875
Ridge: min	1000	0.941 (0.052)	0.958	1.000	-	-	-
	2000	0.821 (0.090)	0.958	1.000	-	-	-
	3000	0.916 (0.063)	1.000	1.000	-	-	-
	4000	0.809 (0.070)	1.000	1.000	-	-	-
	5000	0.689 (0.080)	0.917	1.000	-	-	-

Table A22: MCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.113 (0.061)	0.011	0.017	0.011	0.034	0.057
	2000	0.114 (0.062)	0.028	0.028	0.017	0.045	0.062
	3000	0.094 (0.054)	0.006	0.006	0.000	0.006	0.006
	4000	0.098 (0.057)	0.006	0.006	0.006	0.017	0.006
	5000	0.094 (0.058)	0.000	0.006	0.000	0.011	0.011
Ada Lasso: min	1000	0.103 (0.055)	0.011	0.011	0.011	0.023	0.011
	2000	0.103 (0.058)	0.011	0.011	0.040	0.04	0.006
	3000	0.083 (0.051)	0.000	0.000	0.000	0.006	0.000
	4000	0.090 (0.056)	0.006	0.006	0.006	0.011	0.006
	5000	0.084 (0.056)	0.000	0.000	0.000	0.011	0.028
Ridge: min	1000	0.100 (0.03)	0.068	0.108	-	-	-
	2000	0.048 (0.017)	0.085	0.091	-	-	-
	3000	0.030 (0.012)	0.034	0.108	-	-	-
	4000	0.019 (0.011)	0.040	0.097	-	-	-
	5000	0.008 (0.008)	0.011	0.142	-	-	-

Table A23: MCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.111 (0.069)	0.000	0.023	0.000	0.023	0.023
	2000	0.111 (0.066)	0.011	0.023	0.011	0.040	0.040
	3000	0.116 (0.058)	0.017	0.023	0.017	0.040	0.062
	4000	0.078 (0.041)	0.017	0.017	0.017	0.028	0.017
	5000	0.083 (0.055)	0.000	0.011	0.011	0.017	0.011
Ada Lasso: min	1000	0.101 (0.063)	0.000	0.006	0.000	0.023	0.000
	2000	0.101 (0.059)	0.011	0.011	0.011	0.034	0.057
	3000	0.105 (0.055)	0.023	0.023	0.023	0.023	0.023
	4000	0.069 (0.041)	0.006	0.017	0.017	0.017	0.017
	5000	0.074 (0.053)	0.000	0.006	0.000	0.017	0.023
Ridge: min	1000	0.121 (0.038)	0.068	0.131	-	-	-
	2000	0.050 (0.015)	0.051	0.108	-	-	-
	3000	0.039 (0.013)	0.057	0.159	-	-	-
	4000	0.034 (0.007)	0.051	0.091	-	-	-
	5000	0.025 (0.009)	0.040	0.102	-	-	-

Table A24: MCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$. The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.109 (0.061)	0.006	0.023	0.006	0.051	0.006
	2000	0.087 (0.046)	0.028	0.028	0.028	0.028	0.028
	3000	0.105 (0.058)	0.011	0.011	0.011	0.017	0.011
	4000	0.113 (0.063)	0.006	0.017	0.017	0.034	0.017
	5000	0.089 (0.049)	0.006	0.006	0.006	0.023	0.045
Ada Lasso: min	1000	0.098 (0.056)	0.000	0.023	0.011	0.045	0.000
	2000	0.078 (0.045)	0.023	0.023	0.023	0.023	0.028
	3000	0.096 (0.055)	0.011	0.011	0.011	0.011	0.011
	4000	0.100 (0.057)	0.006	0.006	0.017	0.017	0.017
	5000	0.079 (0.049)	0.006	0.006	0.006	0.006	0.006
Ridge: min	1000	0.092 (0.030)	0.045	0.102	-	-	-
	2000	0.046 (0.013)	0.045	0.102	-	-	-
	3000	0.043 (0.011)	0.057	0.091	-	-	-
	4000	0.035 (0.009)	0.102	0.148	-	-	-
	5000	0.021 (0.008)	0.034	0.097	-	-	-

Table A25: MUNCOR-12 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.896 (0.061)	0.994	0.994	0.994	0.994	0.994
	2000	0.914 (0.059)	1.000	1.000	1.000	0.994	0.972
	3000	0.898 (0.050)	1.000	0.994	1.000	0.983	0.983
	4000	0.950 (0.040)	1.000	1.000	1.000	0.989	1.000
	5000	0.903 (0.062)	1.000	1.000	1.000	1.000	0.977
Ada Lasso: min	1000	0.901 (0.052)	0.994	0.994	0.994	0.994	0.994
	2000	0.920 (0.049)	1.000	1.000	1.000	0.994	0.972
	3000	0.903 (0.043)	1.000	0.994	1.000	0.983	0.983
	4000	0.953 (0.037)	1.000	1.000	1.000	0.994	1.000
	5000	0.910 (0.054)	1.000	1.000	1.000	1.000	0.977
Ridge: min	1000	0.924 (0.043)	0.994	0.983	-	-	-
	2000	0.952 (0.039)	1.000	0.977	-	-	-
	3000	0.918 (0.043)	0.994	0.943	-	-	-
	4000	0.982 (0.026)	1.000	0.960	-	-	-
	5000	0.952 (0.038)	1.000	0.955	-	-	-

Table A26: MUNCOR-12 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.903 (0.062)	1.000	0.994	1.000	0.983	0.972
	2000	0.929 (0.047)	1.000	0.983	1.000	0.983	0.983
	3000	0.969 (0.030)	1.000	1.000	1.000	1.000	1.000
	4000	0.957 (0.036)	1.000	0.994	1.000	0.960	1.000
	5000	0.948 (0.041)	1.000	1.000	1.000	1.000	0.972
Ada Lasso: min	1000	0.910 (0.054)	1.000	0.994	1.000	0.983	0.972
	2000	0.935 (0.044)	1.000	0.994	1.000	0.983	0.983
	3000	0.973 (0.028)	1.000	1.000	1.000	1.000	0.625
	4000	0.960 (0.035)	1.000	0.994	1.000	0.989	1.000
	5000	0.953 (0.039)	1.000	1.000	1.000	1.000	0.972
Ridge: min	1000	0.984 (0.028)	0.994	0.926	-	-	-
	2000	0.993 (0.016)	1.000	0.943	-	-	-
	3000	0.989 (0.020)	1.000	1.000	-	-	-
	4000	0.985 (0.021)	1.000	0.955	-	-	-
	5000	0.982 (0.025)	1.000	0.972	-	-	-

Table A27: MUNCOR-24 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.788 (0.040)	0.667	0.848	0.810	0.848	0.810
	2000	0.774 (0.036)	0.771	0.830	0.830	0.851	0.830
	3000	0.763 (0.035)	0.851	0.887	0.851	0.923	0.851
	4000	0.831 (0.033)	0.896	0.893	0.896	0.890	0.897
	5000	0.862 (0.027)	0.917	0.911	0.917	0.902	0.902
Ada Lasso: min	1000	0.788 (0.041)	0.667	0.851	0.81	0.848	0.810
	2000	0.773 (0.036)	0.771	0.830	0.830	0.851	0.830
	3000	0.762 (0.035)	0.851	0.887	0.851	0.902	0.923
	4000	0.830 (0.033)	0.896	0.896	0.896	0.890	0.897
	5000	0.863 (0.026)	0.917	0.914	0.917	0.902	0.914
Ridge: min	1000	0.678 (0.031)	0.750	0.846	-	-	-
	2000	0.690 (0.036)	0.923	0.921	-	-	-
	3000	0.680 (0.047)	0.833	0.889	-	-	-
	4000	0.689 (0.036)	0.896	0.901	-	-	-
	5000	0.677 (0.038)	0.917	0.885	-	-	-

Table A28: MUNCOR-24 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.83 (0.028)	0.729	0.771	0.729	0.789	0.729
	2000	0.848 (0.029)	0.896	0.896	0.896	0.893	0.918
	3000	0.831 (0.030)	0.875	0.875	0.875	0.858	0.875
	4000	0.846 (0.033)	0.896	0.914	0.896	0.908	0.896
	5000	0.866 (0.027)	0.938	0.953	0.938	0.947	0.947
Ada Lasso: min	1000	0.830 (0.028)	0.729	0.771	0.729	0.771	0.729
	2000	0.849 (0.028)	0.896	0.896	0.896	0.893	0.918
	3000	0.833 (0.029)	0.875	0.875	0.875	0.864	0.875
	4000	0.848 (0.031)	0.875	0.914	0.896	0.908	0.908
	5000	0.867 (0.026)	0.938	0.953	0.938	0.947	0.947
Ridge: min	1000	0.674 (0.040)	0.729	0.856	-	-	-
	2000	0.679 (0.051)	0.896	0.918	-	-	-
	3000	0.744 (0.033)	0.875	0.878	-	-	-
	4000	0.717 (0.033)	0.917	0.898	-	-	-
	5000	0.702 (0.028)	0.955	0.914	-	-	-

Table A29: MCOR-12 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.853 (0.050)	0.875	0.875	0.875	0.894	0.875
	2000	0.878 (0.046)	0.958	0.977	0.958	0.955	0.977
	3000	0.883 (0.049)	1.000	1.000	1.000	0.949	1.000
	4000	0.880 (0.049)	1.000	1.000	1.000	0.989	0.972
	5000	0.887 (0.046)	0.994	0.989	0.994	0.972	0.972
Ada Lasso: min	1000	0.854 (0.042)	0.875	0.875	0.875	0.894	0.875
	2000	0.884 (0.043)	0.958	0.947	0.958	0.977	0.977
	3000	0.886 (0.043)	1.000	1.000	1.000	0.966	1.000
	4000	0.894 (0.042)	1.000	1.000	1.000	0.994	0.972
	5000	0.903 (0.041)	0.994	0.989	0.994	0.972	0.972
Ridge: min	1000	0.866 (0.048)	0.894	0.860	-	-	-
	2000	0.857 (0.040)	0.943	0.943	-	-	-
	3000	0.868 (0.029)	0.989	0.886	-	-	-
	4000	0.923 (0.032)	0.994	0.932	-	-	-
	5000	0.937 (0.024)	0.955	0.955	-	-	-

Table A30: MCOR-12 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.868 (0.045)	0.958	0.983	0.953	0.966	0.958
	2000	0.896 (0.050)	0.994	0.989	0.989	0.955	0.960
	3000	0.877 (0.047)	0.958	0.977	0.958	0.955	0.958
	4000	0.925 (0.044)	1.000	0.977	1.000	0.938	1.000
	5000	0.914 (0.040)	1.000	1.000	1.000	0.966	1.000
Ada Lasso: min	1000	0.876 (0.038)	0.958	0.994	0.958	0.977	0.958
	2000	0.914 (0.042)	0.994	0.994	0.994	0.966	0.994
	3000	0.893 (0.040)	1.000	0.983	0.958	0.960	1.000
	4000	0.937 (0.039)	1.000	0.977	1.000	0.949	1.000
	5000	0.930 (0.036)	1.000	1.000	1.000	0.994	1.000
Ridge: min	1000	0.910 (0.017)	0.900	0.943	-	-	-
	2000	0.956 (0.033)	0.941	0.926	-	-	-
	3000	0.923 (0.033)	0.941	0.938	-	-	-
	4000	0.956 (0.036)	1.000	0.903	-	-	-
	5000	0.969 (0.026)	0.989	0.955	-	-	-

Table A31: MCOR-24 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}50$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.834 (0.038)	0.890	0.887	0.890	0.920	0.909
	2000	0.847 (0.042)	0.923	0.923	0.929	0.915	0.927
	3000	0.835 (0.039)	0.976	0.976	0.854	0.976	0.976
	4000	0.877 (0.036)	0.893	0.914	0.893	0.929	0.893
	5000	0.868 (0.040)	0.917	0.935	0.917	0.973	0.973
Ada Lasso: min	1000	0.822 (0.037)	0.869	0.890	0.869	0.884	0.869
	2000	0.832 (0.040)	0.869	0.932	0.918	0.918	0.810
	3000	0.820 (0.040)	0.854	0.917	0.854	0.976	0.854
	4000	0.868 (0.034)	0.893	0.893	0.893	0.932	0.893
	5000	0.859 (0.037)	0.917	0.917	0.917	0.953	0.965
Ridge: min	1000	0.924 (0.029)	0.966	0.946	-	-	-
	2000	0.914 (0.031)	0.957	0.955	-	-	-
	3000	0.924 (0.027)	0.962	0.946	-	-	-
	4000	0.910 (0.035)	0.938	0.931	-	-	-
	5000	0.892 (0.040)	0.911	0.929	-	-	-

Table A32: MCOR-24 - Variable Selection AUC scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean AUC scores across $M = 500$ fits of each regularised model with $I_R = 1{:}100$. The mean AUC score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean AUC (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: min	1000	0.877 (0.032)	0.750	0.884	0.750	0.926	0.947
	2000	0.902 (0.033)	0.890	0.884	0.890	0.897	0.918
	3000	0.862 (0.035)	0.887	0.884	0.887	0.876	0.906
	4000	0.907 (0.034)	0.908	0.908	0.908	0.902	0.908
	5000	0.886 (0.034)	0.875	0.953	0.953	0.971	0.953
Ada Lasso: min	1000	0.864 (0.029)	0.833	0.851	0.854	0.884	0.750
	2000	0.894 (0.031)	0.890	0.890	0.89	0.879	0.909
	3000	0.858 (0.034)	0.884	0.884	0.884	0.884	0.884
	4000	0.901 (0.033)	0.893	0.887	0.908	0.908	0.908
	5000	0.881 (0.033)	0.875	0.893	0.875	0.95	0.947
Ridge: min	1000	0.887 (0.042)	0.966	0.935	-	-	-
	2000	0.877 (0.045)	0.974	0.946	-	-	-
	3000	0.888 (0.034)	0.951	0.920	-	-	-
	4000	0.936 (0.029)	0.974	0.955	-	-	-
	5000	0.863 (0.029)	0.938	0.949	-	-	-

Table A33: MCOR-24 - Variable Selection TPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean TPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$ and $\lambda =$ "lambda.1se". The mean TPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean TPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: 1se	1000	0.658 (0.091)	0.042	0.708	0.542	0.792	0.542
	2000	0.797 (0.063)	0.208	0.708	0.750	0.792	0.500
	3000	0.896 (0.053)	0.958	0.875	0.958	0.958	0.333
	4000	0.884 (0.044)	0.833	0.833	0.875	0.875	0.292
	5000	0.894 (0.041)	0.875	0.750	0.875	0.875	0.292
Ada Lasso: 1se	1000	0.436 (0.099)	0.042	0.500	0.208	0.542	0.208
	2000	0.571 (0.073)	0.208	0.500	0.750	0.583	0.417
	3000	0.630 (0.072)	0.333	0.458	0.375	0.667	0.333
	4000	0.677 (0.062)	0.250	0.375	0.833	0.750	0.250
	5000	0.675 (0.051)	0.167	0.375	0.708	0.708	0.167
Ridge: 1se	1000	0.711 (0.084)	1.000	1.000	-	-	-
	2000	0.985 (0.027)	1.000	1.000	-	-	-
	3000	0.997 (0.012)	1.000	1.000	-	-	-
	4000	0.999 (0.01)	1.000	1.000	-	-	-
	5000	0.988 (0.027)	1.000	1.000	-	-	-

Table A34: MCOR-24 - Variable Selection FPR scores for each automatic selection method based on the variable ranking and selection algorithm, along with the mean FPR scores across $M = 500$ fits of each regularised model with $I_R = 1{:}1000$ and $\lambda =$ "lambda.1se". The mean FPR score for ridge regression corresponds to cRank applied to an individual fit.

Model	n_b	Mean FPR (s.d)	Automatic Selection Method				
			mRank	cRank	mPdrop	cPdrop	Clustering
Lasso: 1se	1000	0.023 (0.015)	0.000	0.017	0.006	0.017	0.006
	2000	0.018 (0.008)	0.000	0.000	0.006	0.023	0.000
	3000	0.013 (0.007)	0.006	0.006	0.006	0.006	0.000
	4000	0.020 (0.008)	0.006	0.006	0.006	0.006	0.000
	5000	0.011 (0.007)	0.000	0.000	0.000	0.000	0.000
Ada Lasso: 1se	1000	0.007 (0.009)	0.000	0.000	0.000	0.006	0.000
	2000	0.002 (0.003)	0.000	0.000	0.000	0.000	0.000
	3000	0.001 (0.002)	0.000	0.000	0.000	0.000	0.000
	4000	0.002 (0.003)	0.000	0.000	0.006	0.000	0.000
	5000	0.000 (0.001)	0.000	0.000	0.000	0.000	0.000
Ridge: 1se	1000	0.042 (0.009)	0.068	0.085	-	-	-
	2000	0.060 (0.006)	0.068	0.085	-	-	-
	3000	0.062 (0.006)	0.074	0.085	-	-	-
	4000	0.066 (0.005)	0.068	0.108	-	-	-
	5000	0.056 (0.009)	0.068	0.108	-	-	-

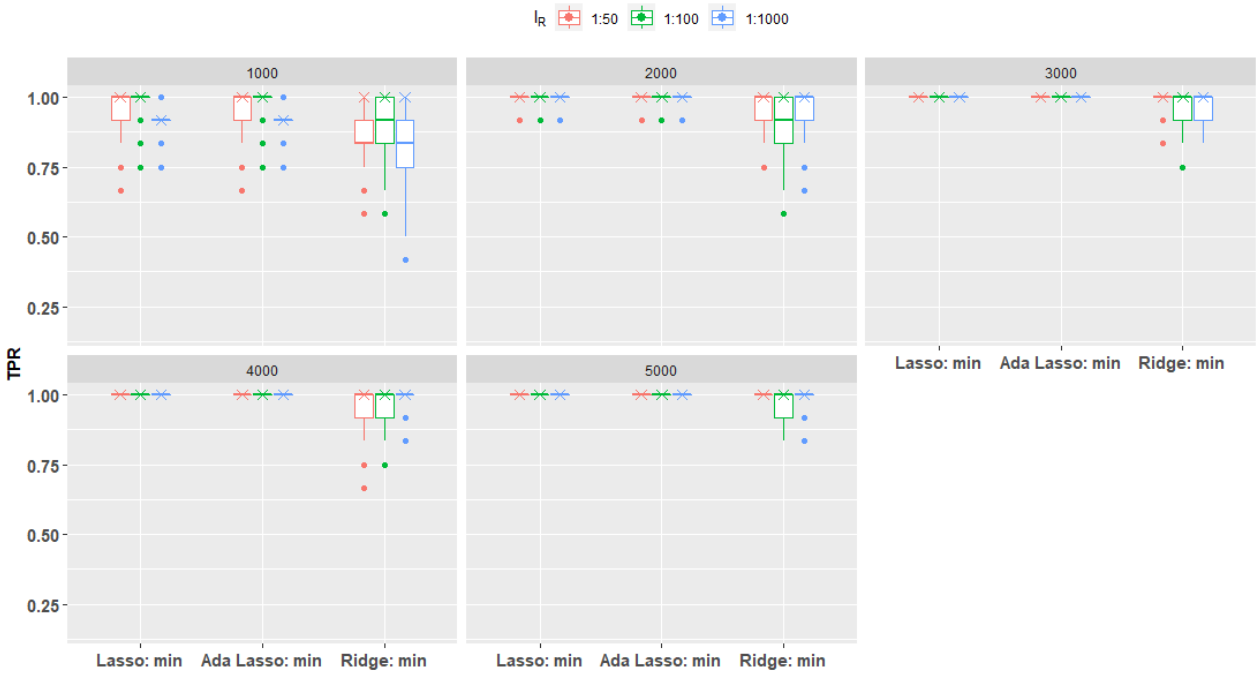


Figure A1: MUNCOR-12 - Distribution of variable selection TPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

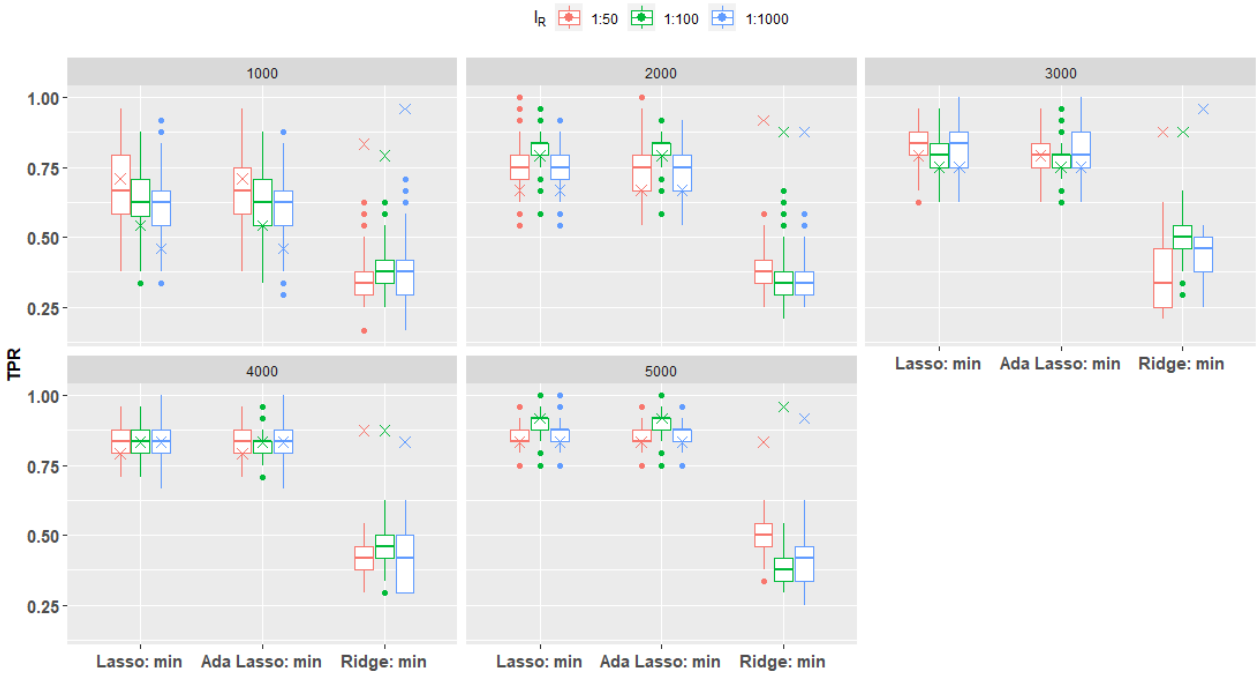


Figure A2: MUNCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Figure A3: MCOR-12 - Distribution of variable selection TPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "×" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

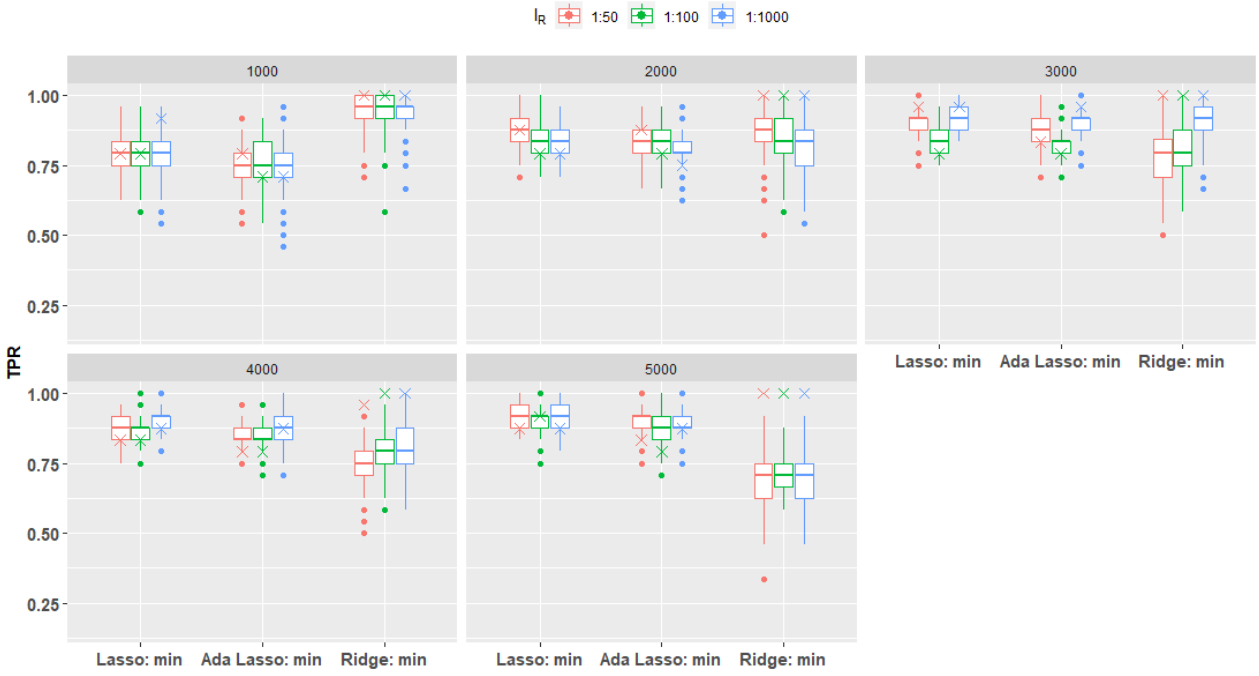


Figure A4: MCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "×" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

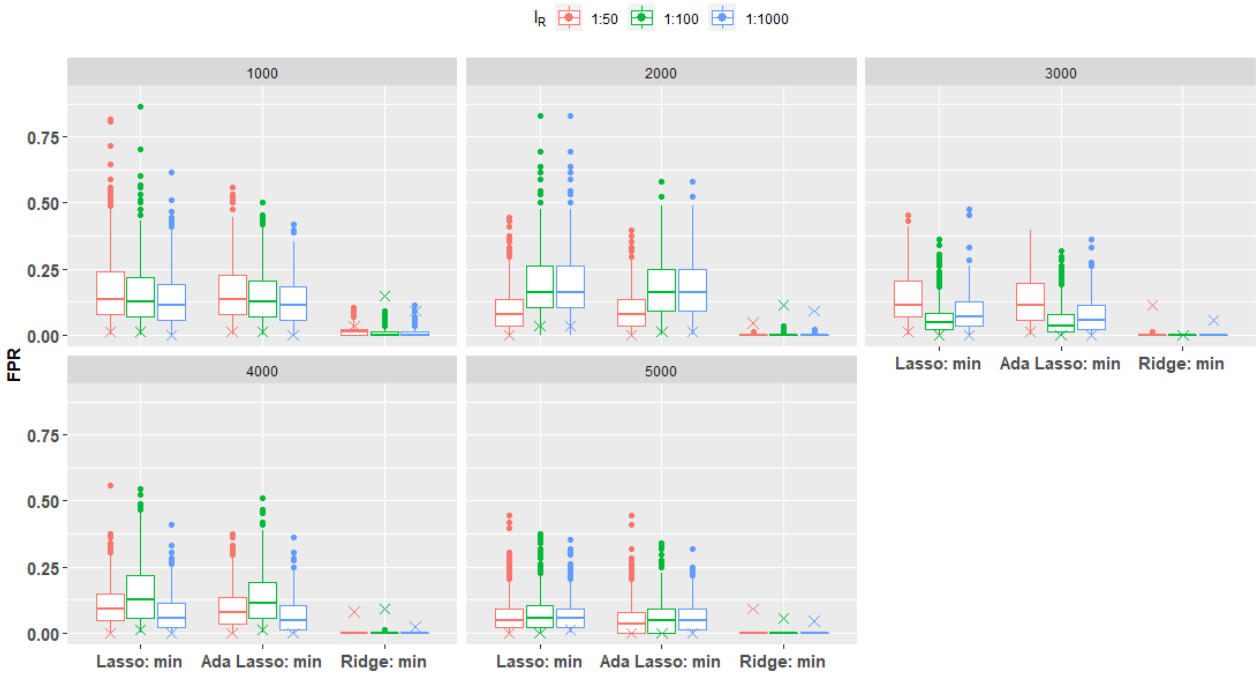


Figure A5: MUNCOR-12 - Distribution of variable selection FPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "×" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

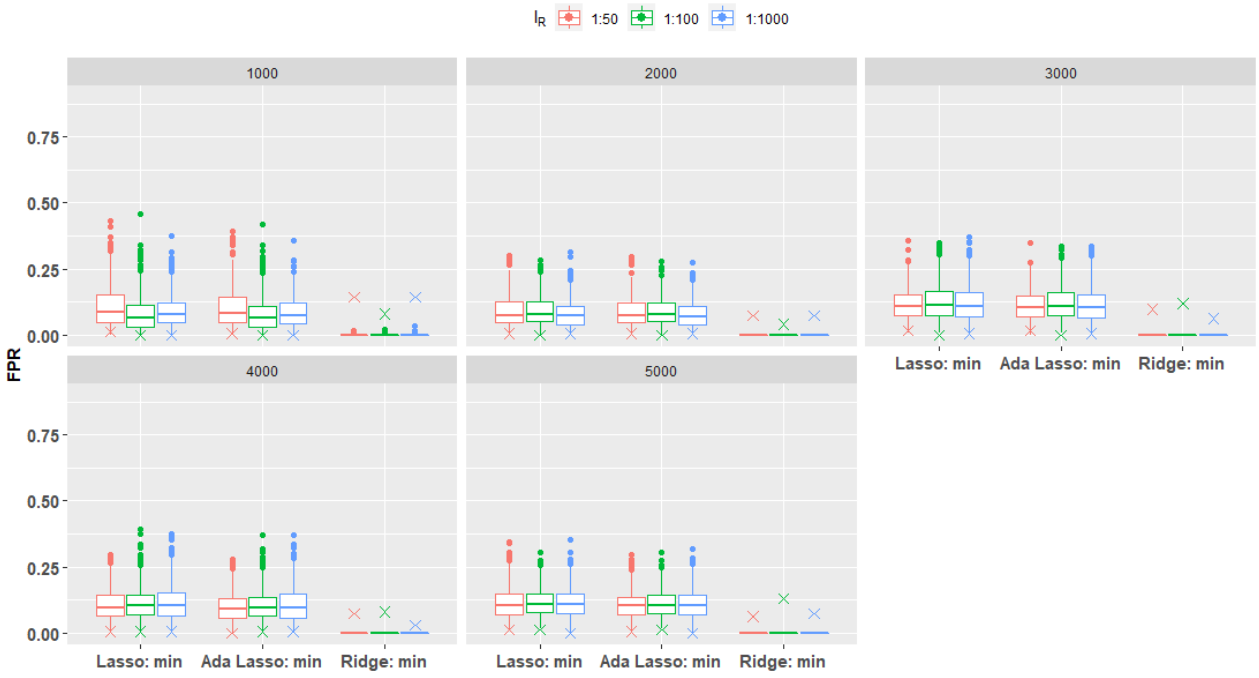


Figure A6: MUNCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

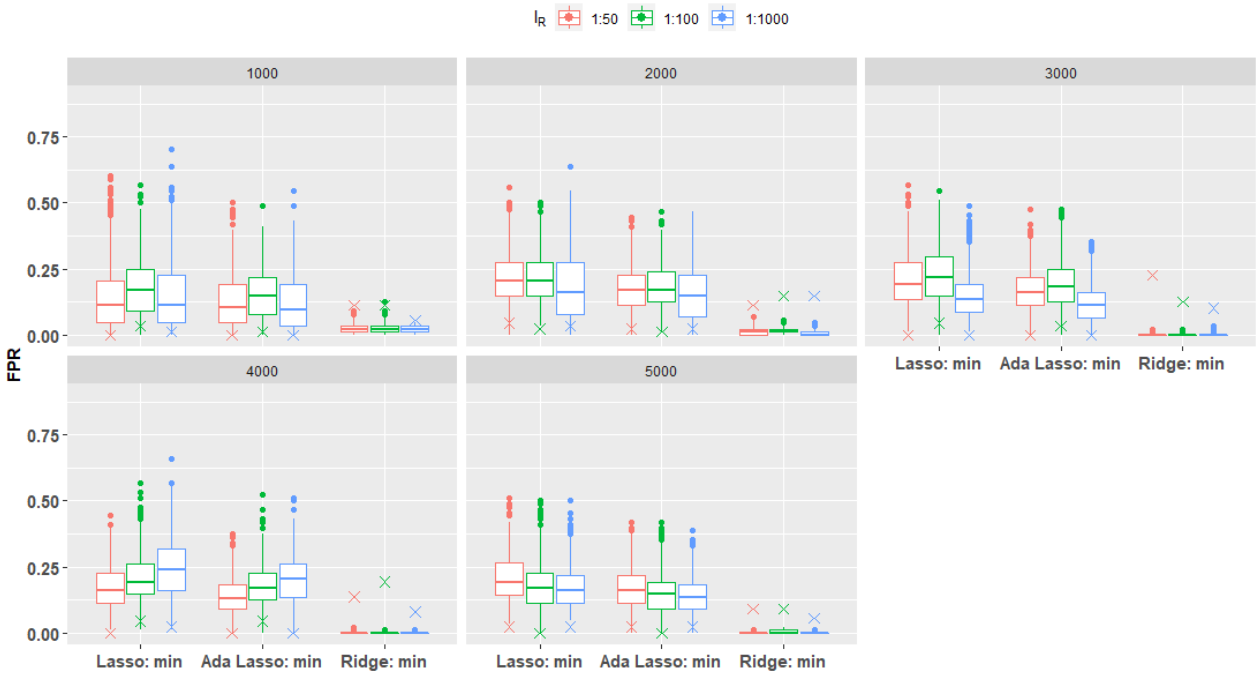


Figure A7: MCOR-12 - Distribution of variable selection FPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "×" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

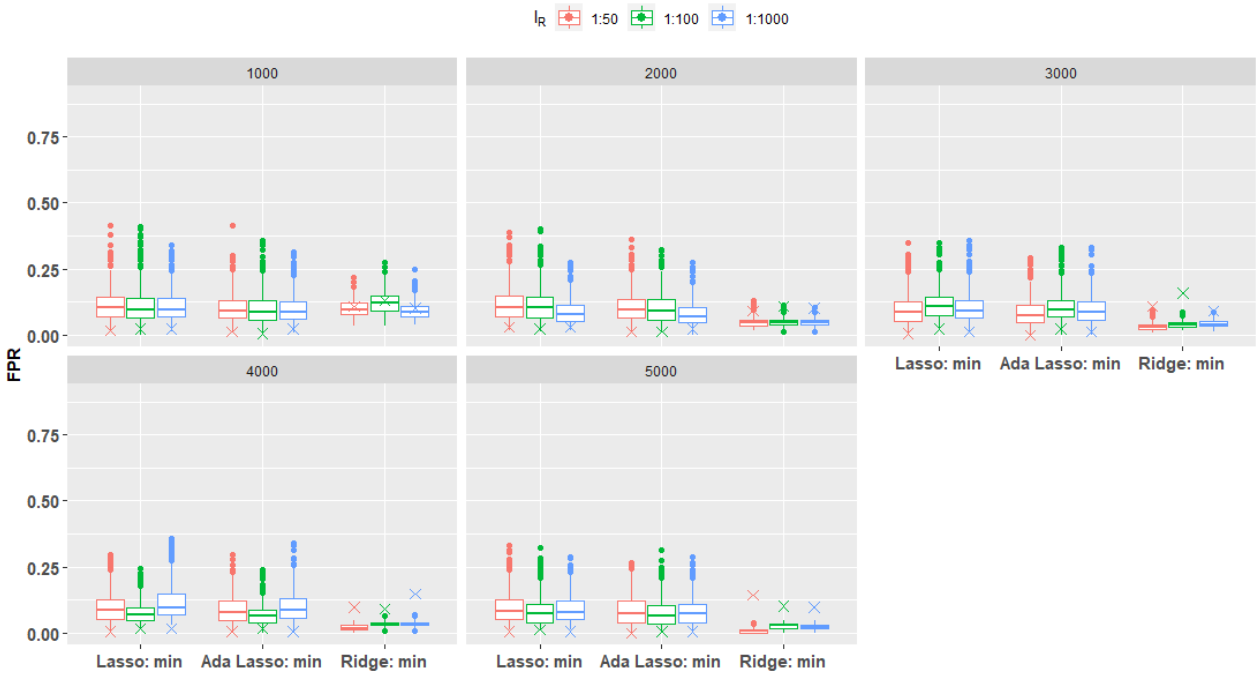


Figure A8: MCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits for each I_R across all n_b's. The marker "×" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

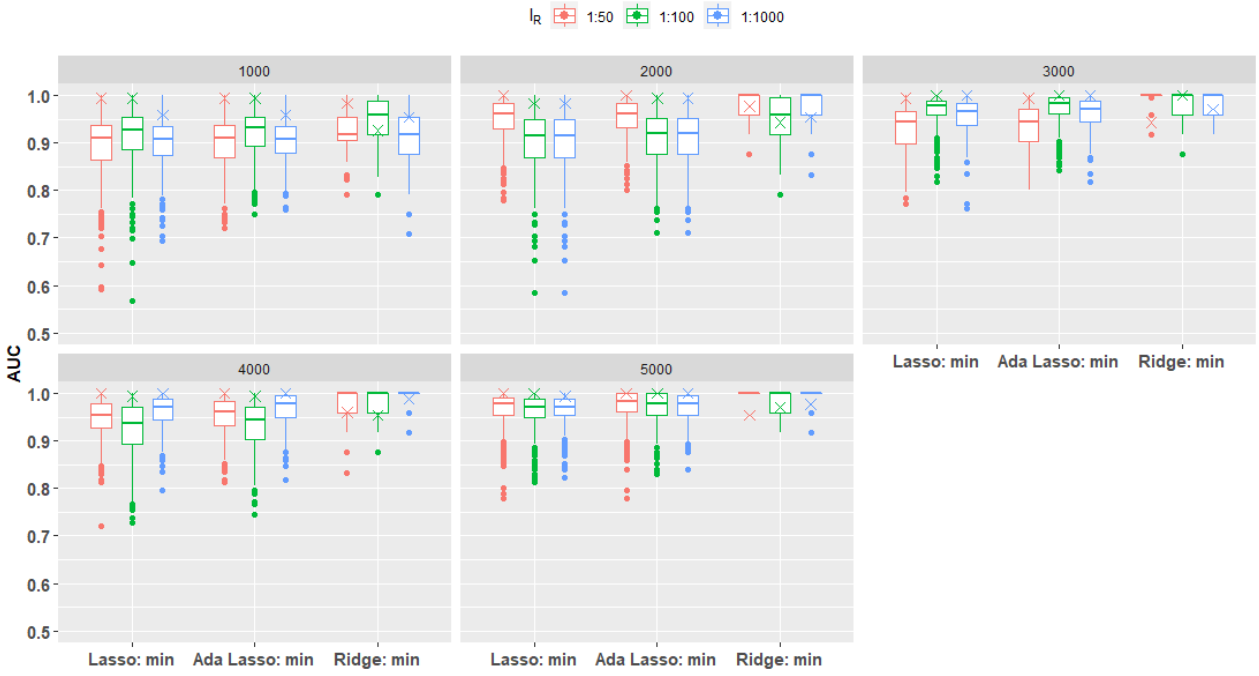


Figure A9: MUNCOR-12 - Distribution of variable selection AUC scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

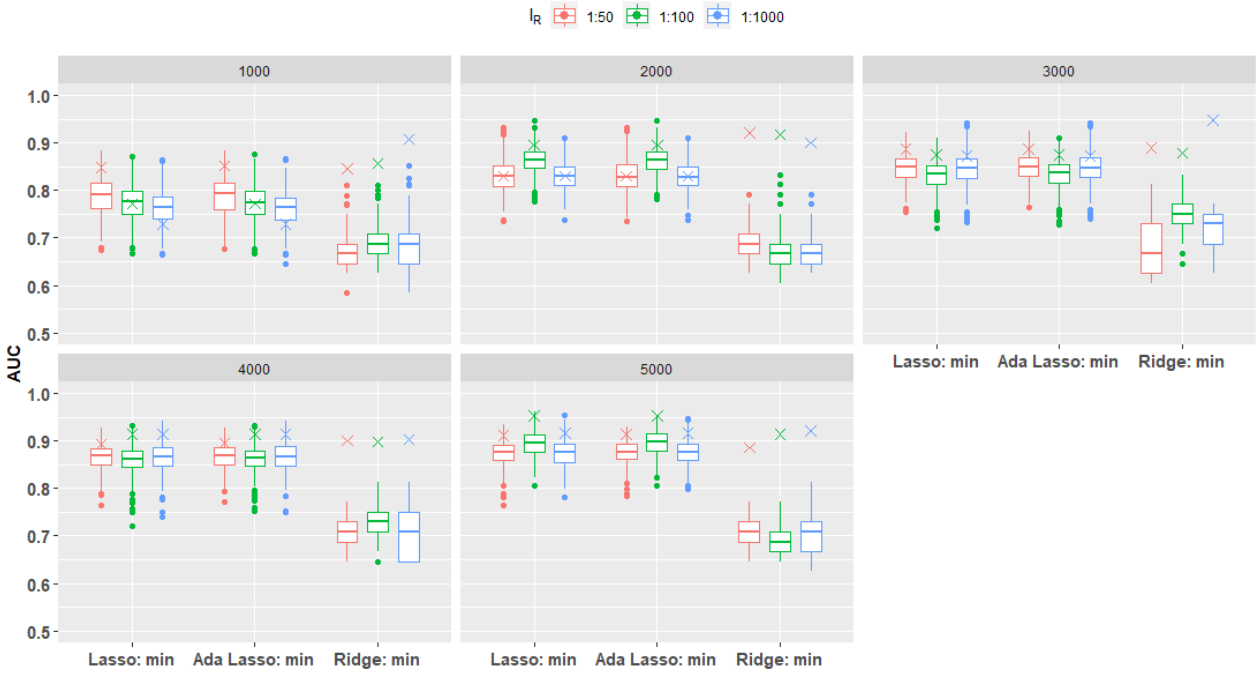


Figure A10: MUNCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

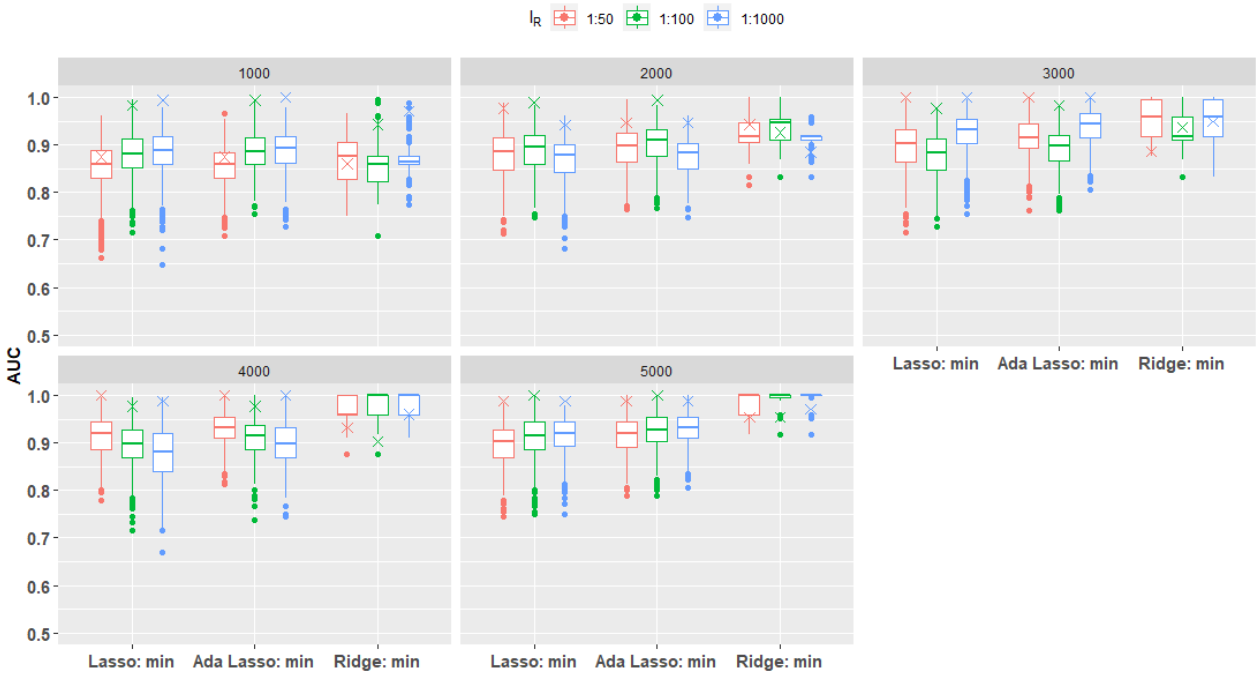


Figure A11: MCOR-12 - Distribution of variable selection AUC scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

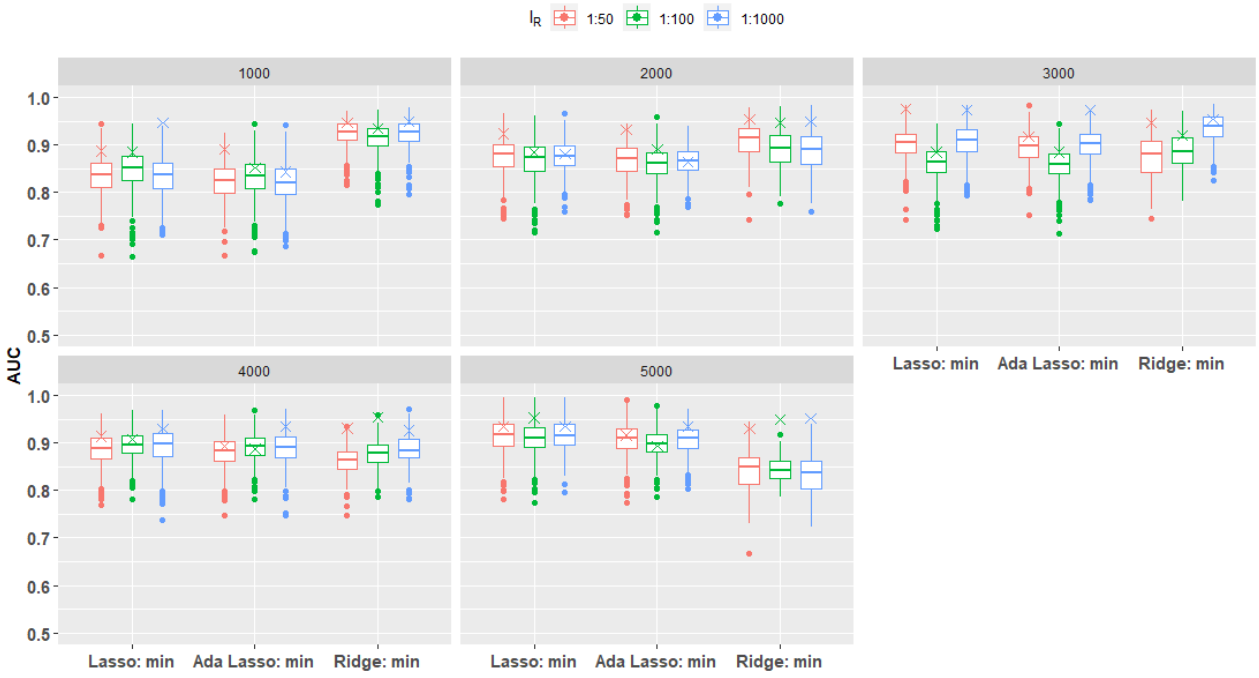


Figure A12: MCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits for each I_R across all n_b's. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

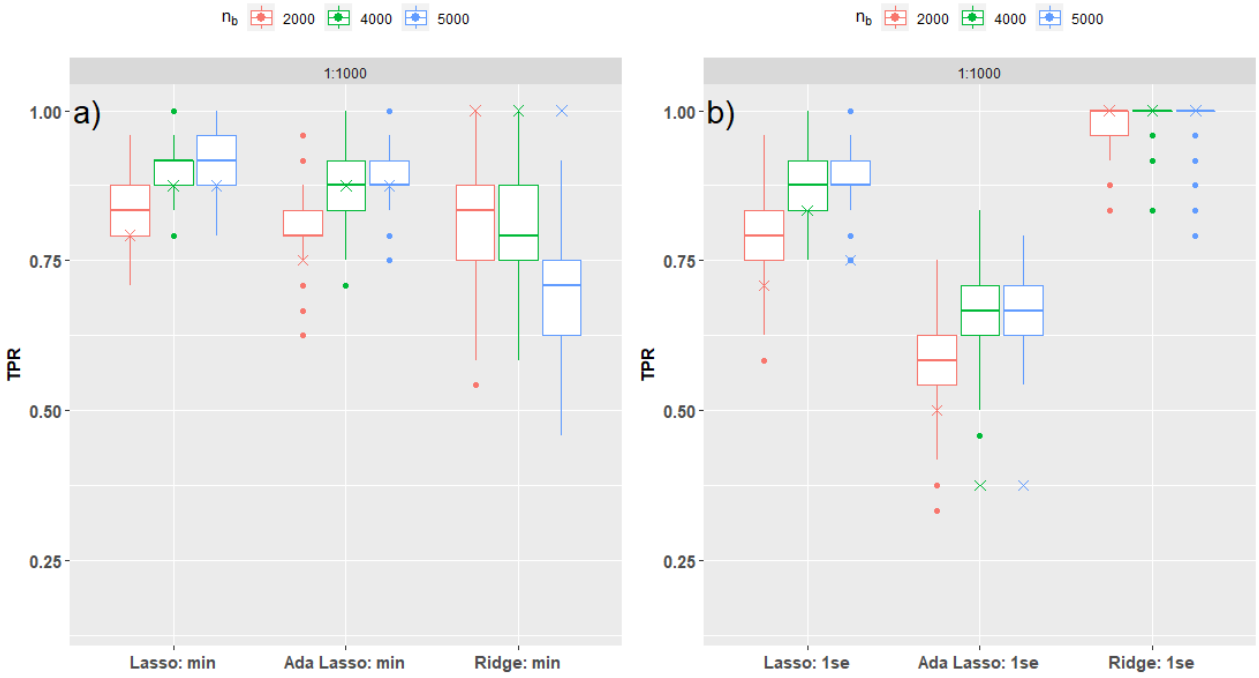


Figure A13: MCOR-24 - Distribution of variable selection TPR scores across $M = 500$ model fits with (a) $\lambda =$ "lambda.min" and (b) $\lambda =$ "lambda.1se" for $n_b = 2000, 4000, 5000$. The marker "$\times$" corresponds to the TPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

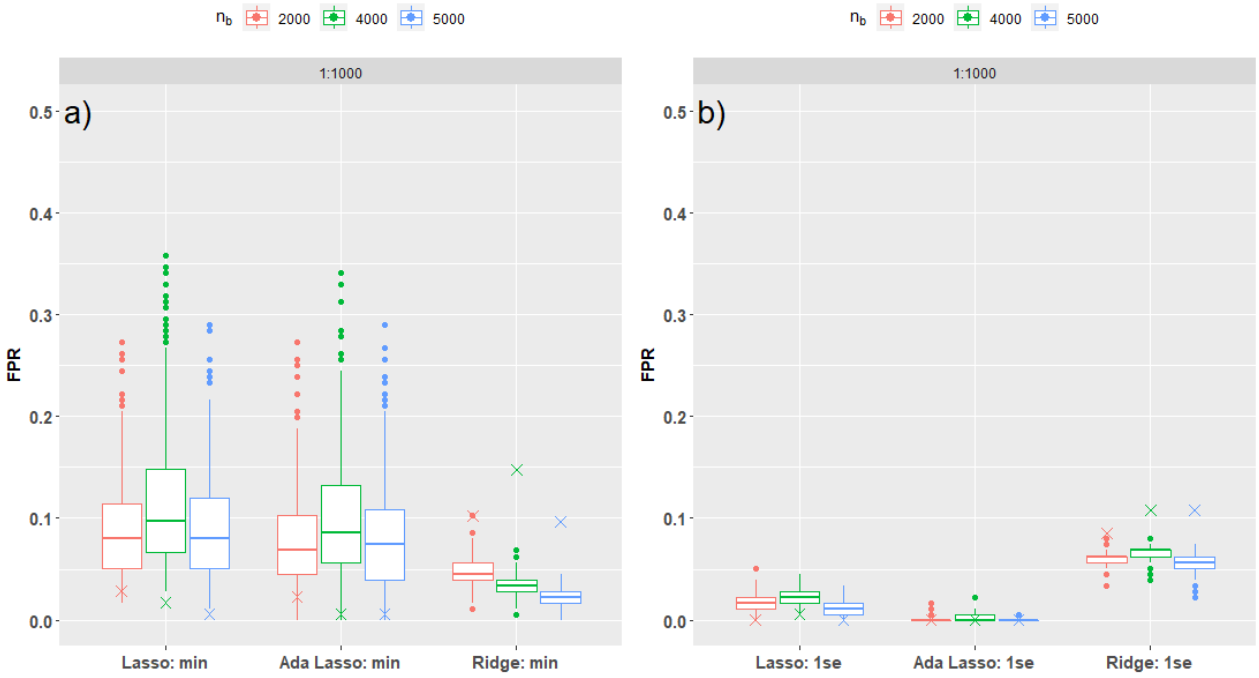


Figure A14: MCOR-24 - Distribution of variable selection FPR scores across $M = 500$ model fits with (a) $\lambda =$ "lambda.min" and (b) $\lambda =$ "lambda.1se" for $n_b = 2000, 4000, 5000$. The marker "$\times$" corresponds to the FPR score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

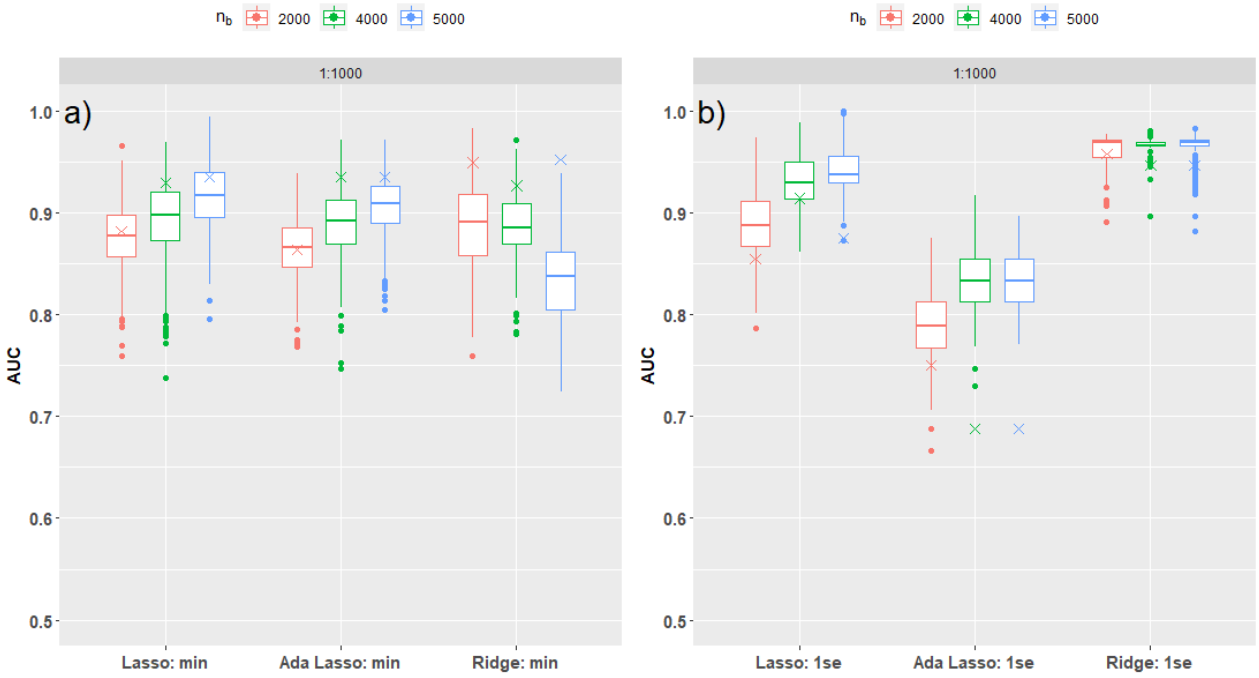


Figure A15: MCOR-24 - Distribution of variable selection AUC scores across $M = 500$ model fits with (a) $\lambda =$ "lambda.min" and (b) $\lambda =$ "lambda.1se" for $n_b = 2000, 4000, 5000$. The marker "$\times$" corresponds to the AUC score obtained using the automatic selection technique cRank from the variable ranking and selection algorithm. The distribution under the ridge regression model corresponds to cRank applied to the individual model fits.

Bibliography

(2021). Forest fires. https://www.nrcan.gc.ca/our-natural-resources/forests/wildland-fires-insects-disturbances/forest-fires/13143. Accessed: 2020-06-28.

Algamal, Z. (2017). An efficient gene selection method for high-dimensional microarray data based on sparse logistic regression. *Electronic Journal of Applied Statistical Analysis*, 10(1):242–256.

Arezzo, M. F. and Guagnano, G. (2018). Response-based sampling for binary choice models with sample selection. *Econometrics*, 6(1):12.

Birgé, L. and Massart, P. (2007). Minimal penalties for gaussian model selection. *Probability Theory and Related Fields*, 138(1-2):33–73.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brillinger, D. R., Preisler, H. K., and Benoit, J. W. (2003). Risk assessment: a forest fire example. *Lecture Notes-Monograph Series*, pages 177–196.

Bruce, D. (1963). How many fires. *Fire Control Notes*, 24(2):45–50.

Cario, M. C. and Nelson, B. L. (1997). Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Citeseer.

Cunningham, A. A. and Martell, D. L. (1973). A stochastic model for the occurrence of man-caused forest fires. *Canadian Journal of Forest Research*, 3(2):282–287.

Ebenuwa, S. H., Sharif, M. S., Alazab, M., and Al-Nemrat, A. (2019). Variance ranking attributes selection techniques for binary classification problem in imbalance data. *IEEE Access*, 7:24649–24666.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499.

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.

Fan, J., Peng, H., et al. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Annals of Statistics*, 32(3):928–961.

Fernández, A., López, V., Galar, M., Del Jesus, M. J., and Herrera, F. (2013). Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97–110.

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

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., and Simon, N. (2021). *glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*. R package version 4.1-1.

García, V., Sánchez, J. S., and Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1):13–21.

García-Pedrajas, N., Pérez-Rodríguez, J., García-Pedrajas, M., Ortiz-Boyer, D., and Fyfe, C. (2012). Class imbalance methods for translation initiation site recognition in dna sequences. *Knowledge-Based Systems*, 25(1):22–34.

Guo, P., Zeng, F., Hu, X., Zhang, D., Zhu, S., Deng, Y., and Hao, Y. (2015). Improved variable selection algorithm using a lasso-type penalty, with an application to assessing hepatitis b infection relevant factors in community residents. *PLoS One*, 10(7):e0134151.

Guyon, X. and Yao, J.-f. (1999). On the underfitting and overfitting sets of models chosen by order selection criteria. *Journal of Multivariate Analysis*, 70(2):221–249.

Hinkley, D. V. (1970). Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1):1–17.

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

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2000). *Applied logistic regression*. Wiley New York.

Jiang, Y., Scott, A. J., and Wild, C. J. (2011). Adjusting for non-response in population-based case-control studies. *International Statistical Review*, 79(2):145–159.

Johnson, R. A., Wichern, D. W., et al. (2002). *Applied multivariate statistical analysis*, volume 5.

Killick, R. and Eckley, I. (2014). changepoint: An r package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19.

Killick, R., Haynes, K., and Eckley, I. (2016). *changepoint: Methods for Changepoint Detection*. R package version 2.2.2.

Kumar, S., Attri, S., and Singh, K. (2019). Comparison of lasso and stepwise regression technique for wheat yield prediction. *Journal of Agrometeorology*, 21(2):188–192.

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal Processing*, 85(8):1501–1510.

López, V., Del Río, S., Benítez, J. M., and Herrera, F. (2015). Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258:5–38.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.0 — For new features, see the 'Changelog' file (in the package source).

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*. Chapman and Hall.

Murray, K. and Conner, M. M. (2009). Methods to quantify variable importance: implications for the analysis of noisy ecological data. *Ecology*, 90(2):348–355.

Nadeem, K., Taylor, S., Woolford, D. G., and Dean, C. (2020). Mesoscale spatiotemporal predictive models of daily human-and lightning-caused wildland fire occurrence in british columbia. *International Journal of Wildland Fire*, 29(1):11–27.

Shi, W., Wahba, G., Irizarry, R. A., Bravo, H. C., and Wright, S. J. (2012). The partitioned lasso-patternsearch algorithm with application to gene expression data. *BMC Bioinformatics*, 13(1):1–10.

Smith, G. (2018). Step away from stepwise. *Journal of Big Data*, 5(1):1–12.

Somers, R. H. (1962). A new asymmetric measure of association for ordinal variables. *American sociological review*, pages 799–811.

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

Touloumis, A. (2016). Simulating correlated binary and multinomial responses under marginal model specification: The simcormultres package. *The R Journal*, 8(2):79.

Touloumis, A. (2019). *SimCorMultRes: Simulates Correlated Multinomial Responses*. R package version 1.7.0.

Wang, H., Lengerich, B. J., Aragam, B., and Xing, E. P. (2019). Precision lasso: accounting for correlations and linear dependencies in high-dimensional genomic data. *Bioinformatics*, 35(7):1181–1187.

Woolford, D., Bellhouse, D., Braun, W., Dean, C. B., Martell, D., and Sun, J. (2011). A spatiotemporal model for people-caused forest fire occurrence in the Romeo Malette forest. *Journal of Environmental Statistics*, 2:2–16.

Wu, Y. (2020). Can't ridge regression perform variable selection? *Technometrics*, pages 1–9.

Xie, Y. and Manski, C. F. (1989). The logit model and response-based samples. *Sociological Methods & Research*, 17(3):283–302.

Yu, H., Sun, C., Yang, X., Zheng, S., Wang, Q., and Xi, X. (2018). Lw-elm: A fast and flexible cost-sensitive learning framework for classifying imbalanced data. *IEEE Access*, 6:28488–28500.

Zhang, C.-X., Zhang, J.-S., and Yin, Q.-Y. (2017). A ranking-based strategy to prune variable selection ensembles. *Knowledge-Based Systems*, 125:13–25.

Zhao, H., Sun, D., Li, G., and Sun, J. (2018). Variable selection for recurrent event data with broken adaptive ridge regression. *Canadian Journal of Statistics*, 46(3):416–428.

Zhou, S., van de Geer, S., and Bühlmann, P. (2009). Adaptive lasso for high dimensional regression and gaussian graphical modeling. *arXiv preprint arXiv:0903.2515*.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.