Expert Systems With Applications 217 (2023) 119535

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Random feature selection using random subspace logistic regression

Nuttanan Wichitaksorn [*], Yingyue Kang, Faqiang Zhang

Department of Mathematical Sciences, Auckland University of Technology, Private Bag 92006, Auckland 1142, New Zealand

ARTICLE INFO

JEL classification:
C15
C25
C55

Keywords:
Feature selection
Logistic regression
Lasso logistic
Random subspace
Bootstrap

ABSTRACT

Feature selection becomes a prominent method in the big data era. The logistic regression model is a wrapper method that provides better classification or prediction accuracy but it is computationally expensive. In this study, we propose the random subspace logistic regression where features are randomly selected through bootstrap cycles. The random subspace regression method is applied to both standard and lasso logistic regression models. Using the simulated and empirical data, our proposed random subspace logistic regression shows favorable results and can be a promising alternative for flat feature selection.

1. Introduction

It is generally known that big data can contain three characters including volume, variety, and velocity, which are the main cause of the complexity. However, these can basically be summarized into two components: observations and variables. With the increasing computer performance, large observations are easier to deal with while a large number of variables makes data analysis more complicated (Li & Liu, 2017). Hence, feature selection was invented and designed to overcome this issue.

Feature selection becomes a prominent method in choosing relevant variables that improve the model performance, e.g., classification or predictability. Precisely, feature selection is useful to reduce the dimensions or the number of irrelevant variables, and that can remove some noises or errors. Following that, the growing literature on feature selection reflects its popularity and significance for better accuracy in classification in many areas ranging from environmental and health sciences to finance, see, among others, an exhaustive list (Arauzo-Azofra et al., 2011; Chen et al., 2020; Guyon & Elisseeff, 2003; Hira & Gillies, 2015; Miao & Niu, 2016) and references therein.

Recently, many studies focus on new feature selection methods or algorithms for better classification and identification of relevant and irrelevant features that enhance the computational efficiency (Khaire & Dhanalakshmi, 2019). In this study, we focus on the selection of flat features or the independent variables that are relevant or influential to the model. Generally, there are three types of feature selection algorithms including wrapper, filter, and embedded methods. Though different methods have different advantages and disadvantages, the wrapper method seems to be the best one in terms of classification or prediction accuracy (see Table A.1 in the appendix for our recent comparison of different classification methods) but many of them are not computationally efficient.

Logistic regression (also known as logit) is one of the wrapper methods that has been widely used for categorical data in many applications, (e.g., Cheng et al., 2006; Isachenko & Strijov, 2018; Khandezamin et al., 2020; Weber et al., 2004). The logistic regression was also improved to include the L_1 and L_2 regularizations (Ng, 2004). This can be considered as an embedded or hybrid method where the wrapper and filter methods are combined. In addition, there was a development to reduce the computational burden of the logistic regression for the high-dimensional datasets where the features were selected in parallel (Singh et al., 2009).

Even with its decent performance in the classification, logistic regression still has a shortcoming, which is computational inefficiency. This is generally due to its usual estimation method through an optimization from the hill-climbing Newton family such as Newton–Raphson and its modifications and extensions. Hence, dealing with a large set of features seems difficult for the standard logistic regression.

In this study, we propose a novel method to overcome the problem by randomly selecting the features through Monte Carlo (bootstrap) simulations for the logistic regression. Precisely, we base our method on the concept of random subspace regression (Boot & Nibbering, 2019). The advantage of this method is to allow us to use logistic regression for large datasets with high-dimensional features. We assess our proposed random subspace logistic regression through two large

[*] Corresponding author.
E-mail addresses: nuttanan.wichitaksorn@aut.ac.nz (N. Wichitaksorn), tsuki040810@gmail.com (Y. Kang), faqiang.zhang@uq.net.au (F. Zhang).

https://doi.org/10.1016/j.eswa.2023.119535
Received 14 April 2022; Received in revised form 8 January 2023; Accepted 8 January 2023
Available online 13 January 2023

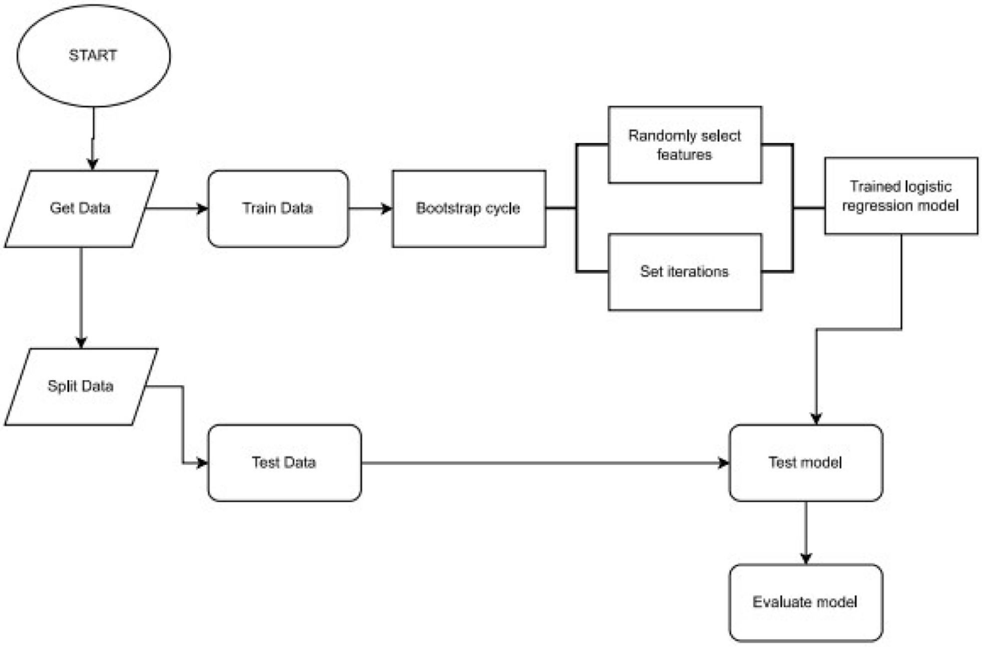

Fig. 1. Random Subspace logistic regression workflow.

datasets from the UCI Machine Learning Repository and Kaggle and compare its performance with other competing methods including standard logistic regression, lasso, random forest, neural network, and support vector machine (SVM). We found that our proposed method can be a promising alternative.

Our results from two empirical studies with large datasets show that, compared to the standard logistic regression, the proposed random subspace method returns lower computational time and higher prediction accuracy. In many cases, increasing the number of bootstrap cycles also raises the accuracy while the low computational time is maintained. However, this might not be the case for the lasso in which increasing the bootstrap cycles leads to higher computational time.

It is worth emphasizing here that our contribution is to facilitate the model estimation for the logistic regression through a random subspace method. To our knowledge, we found a limited extension on this as it has not been widely used. The concept of random subspace regression is quite recent and can be applied to regularizations, as our study shows in the case of lasso. In addition, a theoretical justification is provided while the results from simulation studies prove and confirm the satisfactory performance of our proposed method.

Note that the random subspace logistic regression we propose in this study can be considered as a subset selection method. However, in our study, the subsets of features are randomly selected and iterated using bootstrap cycles. Then, the predicted response values from each subset are aggregated/averaged and evaluated for prediction accuracy. Fig. 1 below summarizes the implementation steps of the proposed random subspace logistic regression.

The rest of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 explains the models and method. Section 4 presents the results. Section 5 provides the concluding remarks and discuss future work.

2. Related work

Features can be categorized as flat, structured, and streaming features. Different methods are used to select different types of features. For example, filter, wrapper, and embedded methods are designed for flat features. Varied penalized classification/regression methods are used for structured data while grafting, alpha-investing and online streaming algorithms are aimed at the streaming features. In this study, we focus only on the flat features as our main purpose is to randomly choose the subsets of independent variables to predict a response variable.

2.1. Selection of flat features

Generally, the selection of flat features has four components including (1) subset generation, (2) evaluation function, (3) iteration stopping criteria, and (4) validation process. As mentioned in Section 1, the methods to select flat features are filter, wrapper, and embedded. Filter methods first evaluate features independently by certain criteria, then select the highly evaluated features (Dash & Liu, 2000; Hall, 2000; Michalak & Kwasnicka, 2006; Yu & Liu, 2003). Criteria for filter methods can be further categorized as (1) similarity-based, (2) information theory-based, and (3) statistical-based (Li et al., 2017).

The similarity-based criteria aim to evaluate the similarity between alternative features where the representative similarity-based criteria include Fisher score (Gu et al., 2011; Hart et al., 2000), Laplacian score (He et al., 2005), and RelieF (Kira & Rendell, 1992; Kononenko et al., 1997; Robnik-Šikonja & Kononenko, 2003). For the information theory-based criteria, the information entropy is evaluated and maximized to reduce the feature redundancy. The most representative information theory-based criterion is the mutual information gain, which measures the relevance of the feature to the target value (Lewis, 1992). The advantages of mutual information gain are their accessible interpretation and computational efficiency. There are also other information theory-based criteria including joint mutual information, conditional mutual information maximization, minimum redundancy maximum relevance, see more details (Brown et al., 2012; Fleuret, 2004; Meyer et al., 2008; Peng et al., 2005; Yang & Moody, 1999). However, the disadvantage of the theory-based criteria is that most of them are only suitable for discrete variables. Lastly, the statistical-based criteria include data variance, t-score, Chi-square score, and Gini score. The data variance independently evaluates the features by their variation with meaningful interpretation. Precisely, information about a feature is contained in its variance or fluctuation. That means the larger the variance is, the more information the feature is generally expected to have.

The main advantage of the filter methods is that it does not require any training with presupposed learning models, hence, the algorithms are more computationally efficient. However, the disadvantage of the filter methods is the lower prediction accuracy compared with wrapper and embedded methods, which take learning models into account while selecting features. In addition, since the filter methods evaluate the features independently, they cannot solve the feature redundancy problem.

As opposed to the filter methods, the wrapper methods evaluate and select the features, based on the performance of presupposed learning models, by subsets (Guan et al., 2004; Hsu et al., 2002; Michalak & Kwasnicka, 2006). In general, the data and learning models both contain biases. However, the wrapper methods take into account the model biases during the subset evaluation, hence, the eventual prediction performance is generally better than the filter methods. However, the computational burden of the wrapper is a lot heavier than that of the filter methods. Given m features, the total number of the feature subsets is 2^m. Hence, it is an NP-hard problem to find the optimal subset within all the subsets by the exhaustive search strategy (Bins & Draper, 2001).

To reduce the time complexity, researchers developed several induction algorithms. Popular induction algorithms include hill-climbing, best-first, greedy algorithms, etc (Guyon & Elisseeff, 2003). The hill-climbing algorithm, e.g., Newton–Raphson, keeps expanding the current subset until the expanding does not improve the prediction performance (Davis, 1991). The best-first algorithm is a sort of heuristic algorithm. The algorithm evaluates the promises of the subsets by a heuristic function and then selects the most promising subset. In this study, we focus on the hill-climbing algorithm used to estimate the logistic regression model for classification purposes. Precisely, we use the wrapper method for feature selection with the expectation that it will return higher accuracy for the classification.

As mentioned above, the filter and wrapper methods both have shortcomings-the filter methods do not contain the interaction with learning models, while the wrapper methods are more computationally expensive. The embedded methods were developed to combine the advantages of the filter and wrapper methods. That is, they have presupposed learning models that provide robust prediction performance, meanwhile, they are a lot more computationally efficient than those of the wrapper methods (Liu & Yu, 2005; Ma & Huang, 2008; Saeys et al., 2007). The embedded methods can be categorized as artificial pruning models and built-in pruning models (Tang et al., 2014). The artificial pruning models acquire coefficients by training original learning models and then artificially shrink some of the coefficients to 0 (Guyon et al., 2002; Mao et al., 1994). In contrast, the built-in pruning models can eliminate some of the features by regularization or splitting criteria. Specifically, the regularization models exert penalties on the coefficients so that some coefficients shrink to 0, such as lasso and adaptive lasso (James et al., 2013; Zou, 2006), while some popular tree splitting criteria exert penalties on the splitting so that some features are not used in the output models, such as ID3 and C4.5 (Quinlan, 1986, 2014). To reap the benefits of the embedded methods, we apply the lasso to the logistic regression and use it in our study.

2.2. Logistic regression and feature selection

In data science, the logistic regression is a classification method that is assumed to be earliest put forward by Joseph Berkson in 1944 (Berkson, 1944). Nowadays, the logistic regression has been widely applied in many fields, such as clinical medicine, remote sensing data recognition, and credit scoring (Ayalew & Yamagishi, 2005; Bensic et al., 2005; Chadwick et al., 2006; Eftekhar et al., 2005; Lee, 2005), among others. The logistic regression is a statistical model with an assumption that the variables are independently and identically distributed. In the high-dimensional datasets, many variables or features could be irrelevant or redundant to the target or response variable. This has posed a challenge to machine learning researchers (Bolón-Canedo et al., 2015; Li & Liu, 2017). Specifically, the mean number of the features in datasets of the UCI machine learning repository are found to have increased over time. In addition to the data irrelevance and redundancy, the increase in computational complexity is also a problem brought with the high dimensionality of data. Feature selection methods can help relieve these two problems. Precisely, in this study we propose a random subspace logistic regression, including the lasso counterpart, by selecting features randomly through Monte Carlo simulations.

2.3. Random subspace regression

Subset feature selection has been widely used and integrated as a part of all methods for flat features. Statistically speaking, subset selection can be regarded as a sampling method. Hence, subset selection and random subspace regression share similar ideas where the latter can be regarded as a subset of the former. When applied to the feature selection, the random subspace regression can be called bagging or feature bagging where the predicted values from each subset of features, not observations, are aggregated through a random sampling method with replacement such as bootstrap (Boot & Nibbering, 2019). The benefit of this random sampling method is to speed up the analysis of big data, e.g., large set of features (Ng, 2017). We expect this random sampling through Monte Carlo simulations, e.g., bootstrap, will reduce the computational burden for the wrapper method or the logistic regression model estimation in our case. In addition, logistic regression tends to perform well with smaller datasets (Varian, 2014). Hence, using the random subspace regression technique might help improve the feature classification.

Note that the random subspace method has also been exploited in some existing works. It can help improve the classification accuracy in the decision forests and regression trees, see Ho (1998) and Pham et al. (2018), among others. Obviously, feature selection, e.g., semi-supervised and unsupervised, is an area where the random subspace has been applied (Huang et al., 2019; Ren et al., 2011). In addition, it can also be embedded in the SVM and the multivariate feature selection (Bertoni et al., 2005; Lai et al., 2006). Bankruptcy and credit risk predictions are the applications where the method has been used (Li et al., 2011; Wang & Ma, 2011). However, the use of random subspace logistic regression in the random feature selection is rarely observed, hence, we chose to implement it in this study.

3. Models and method

3.1. Models

Let y denote the observed binary response variable taking the value either 0 or 1. With y^* as the unobserved response variable, the logistic regression model is given by

$$y_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i,$$

where $i = 1, \dots, N$ is the observation index, $\mathbf{x}_i$ is the $k \times 1$ vector of regressors or features, $'$ denotes the transposition, $\boldsymbol{\beta}$ is the $k \times 1$ vector of regression coefficients, and ϵ_i is the standardized random error following the logistic distribution with mean zero and variance $\pi^2/3$. The relationship between the observed y and the unobserved y^* is

$$y_i = 1 \text{ if } y_i^* > c$$
$$y_i = 0 \text{ if } y_i^* \leq c,$$

where c is a certain threshold, which is 0 in many applications. Based on the logistic distribution assumption of ϵ_i, it follows that

$$\text{Prob}(y_i = 1|\mathbf{x}_i) = \frac{\exp(\mathbf{x}_i'\boldsymbol{\beta})}{1 + \exp(\mathbf{x}_i'\boldsymbol{\beta})}.$$

In the estimation of the logistic regression model, we use the Newton–Raphson hill-climbing method to obtain the parameter estimates, i.e., regression coefficients $\hat{\boldsymbol{\beta}}$. However, with the large set of features $\mathbf{x}_i$, the method does not converge well and results in peculiar parameter values. Hence, estimating the model using a subset of features is a viable alternative.

Let z_i denote $\text{Prob}(y_i = 1|\mathbf{x}_i)$. It follows that the estimated probability $\hat{z}_i = \exp(\mathbf{x}_i'\hat{\boldsymbol{\beta}})/(1+\exp(\mathbf{x}_i'\hat{\boldsymbol{\beta}}))$. In the prediction, we set $\hat{y}_i = 1$ if $\hat{z}_i > q$, otherwise $\hat{y}_i = 0$ where q is the incidence rate or the proportion of $y_i = 0$ in the training set. We can then assess the prediction accuracy between y_i and $\hat{y}_i$.

In addition to the standard logistic regression, we also apply the random subspace regression to the L_1 regularized logistic regression as that in Ng (2004) or so-called lasso logistic where we expect it to return better prediction accuracy. The random subspace lasso logistic regression can significantly reduce the number of irrelevant features.

3.2. Method

The idea behind random subspace logistic regression in this study is to randomly sample features and estimate the corresponding regression coefficients that are used for prediction. The logistic regression is one of the wrapper methods known for its decent predictability performance but computationally expensive. The random subspace logistic regression is aimed to reduce the computational burden while the predictability performance is maintained. Precisely, the random subspace logistic regression uses the bootstrap, which is a Monte Carlo method (here we use these two terms, bootstrap and Monte Carlo, interchangeably) that samples subsets of observations, to get subsets of features. Resulting from this, the predicted response values are aggregated across bootstrap samples and expected to return better predictions. This method is called bagging (Varian, 2014).

For each observation i and bootstrap cycle $j = 1, \ldots, J$, we have the subset of size $s_j < k$ for the features $\mathbf{x}_i^{s_j}$ where s is the pre-determined value for the number of random features in the subset that will be drawn, which is the same across j. Though the s value is the same across j, the random subset $\mathbf{x}^{s_j}$ may not be the same. Hence, we put the subscript j into s to emphasize this difference. For example, let s equal 10. That means, for each bootstrap cycle j, the new random subset $\mathbf{x}^{s_j}$ of size 10 will be drawn while the s_j value is maintained. Note that in our study, each $\mathbf{x}_i^{s_j}$ contains the constant term. Subsequently, we obtain the regression coefficients $\boldsymbol{\beta}^{s_j}$ and the predicted (or estimated) probabilities $\hat{z}_i^{s_j}$. The J values of $\hat{z}_i^{s_j}$ are then aggregated across j to yield $\hat{z}_i$.

With the random sampling of features that results in $\hat{z}_i^{s_j}$ from all J bootstrap cycles, we can apply the consistency theorem of the sample mean from the large-sample distribution theory, which states that "*the mean of a random sample from any population with mean μ and finite variance σ^2 is a consistent estimator of μ*". In our case, for all J values, $E[\hat{z}_i^{s_j}] = \hat{z}_i$ where $plim\ \hat{z}_i^{s_j} = \hat{z}_i$.

The algorithm below shows how we obtain the estimates from the random subspace logistic regression. For each bootstrap cycle j with the pre-determined s,

> *Step 1*: For all observations N, randomly select the subset of features $\mathbf{x}^{s_j}$ and estimate the logistic regression model to obtain $\boldsymbol{\beta}^{s_j}$.
>
> *Step 2*: For each i, calculate and collect $\hat{z}_i^{s_j}$.

Then, for each i, the $\hat{z}_i^{s_j}$ are aggregated across j to get $\hat{z}_i$, which we use to obtain the predicted $\hat{y}_i$, as described in Section 3.1, for accuracy assessment. We perform similar steps for the lasso logistic regression model where the variable selection through penalization occurred in Step 1.

To aggregate the predicted probabilities $\hat{z}_i^{s_j}$, we use the simple and weighted averages. For the weighted average, the deviance function from the logistic regression model evaluated at each j is transformed and used as the weight. However, we found the aggregation using the weighted average, not reported here for conciseness, returns a slightly better prediction accuracy but is not always the case. This is an area we need to explore further but leave it for future research.

Table 1
Prediction accuracy results from simulated data (Unit: %).

No. of	No. of	Model			
cycles	features	Logit (All)	Lasso (All)	Logit (Subspace)	Lasso (Subspace)
50	50	84.5	**85.5**	80.5	80.5
100	50	83.0	**83.5**	81.0	**83.5**
500	50	**87.0**	86.5	82.0	83.0
50	100	**83.0**	82.0	82.5	82.0
100	100	**84.0**	83.5	80.0	80.0
500	100	87.0	88.0	**88.5**	84.5

Note: Bold font indicates the best prediction accuracy.

In addition to the standard and lasso logistic regression models using the random subspace method, we also implement other models and methods including random forest, neural network, and SVM for comparison purposes. After obtaining all results, we use mean absolute deviation to assess the prediction accuracy across models and methods.

4. Results

4.1. Simulated data

We use simulated data to preliminary assess our proposed models and method. In the simulation study, we generate both features and regression coefficients using a standard normal distribution with $k = 201$, to include a constant term, while the error term is generated following the description shown in Section 3.1 with $N = 1{,}000$. Once we obtain the unobserved variable y_i^*, it is then converted to y_i^* using the threshold $c = 0$. The number of observations and features in the simulation study seems not very large but they are sufficient to assess the models and method, including full logistic regression and random subspace logistic and lasso logistic regression models, for their prediction accuracy. We only make a comparison across logistic regression models because the true model that generates the data is the logistic one. Including other models and methods in this context will end up with their inferior results and will be misleading.

The data are split into the training and test sets where the training accounts for 80% of all observations. Though the test data are randomly selected, we also maintain the incidence rate as that of the whole dataset. As this is an experiment and to ensure our algorithm is correct, we chose to write the code and implement it using `MATLAB`. For the lasso, the value for the penalty coefficient λ is automatically searched by `MATLAB` with the cross validation set at 3-fold.

With the incidence rate around 0.5, we implement the models with a different combination of bootstrap cycles (50, 100, and 500) and the number of features (50 and 100). For each combination, we generate different datasets for a fair comparison. Table 1 shows the prediction accuracy across models from the simulation study. Not surprisingly, the full logistic regression model returns a decent prediction accuracy ranging from 83%–87% as it is the true model where all features are included. The model that also performs well (83.5 and 85.5%) is the full lasso logistic regression. Though not the best in many cases, both of the random subspace models show promising results where the lasso is marginally better than the standard one.

Generally, for the random subspace logistic and lasso logistic models, when the number of bootstrap cycles and features increases, their predictability performance is better. However, this cannot be seen clearly from the simulation study results because the new dataset is generated every time the combination changes. Precisely, the dataset from each combination can be treated as a different dataset. That means models can perform differently depending on the dataset. This can be confirmed with the empirical datasets.

Table 2
Results from TUANDROMD dataset.

Model/Method	Accuracy (%)	Comp. time (mins)
Logit (All)	97.54	1.018
Lasso (All)	97.98	1.019
Random Forest	96.52	1.020
Neural Network	98.32	1.016
SVM	96.75	1.020
Logit (Subspace)		
10 cycles with 10 features	82.96	1.017
10 cycles with 50 features	96.75	1.016
10 cycles with 100 features	98.32	1.016
10 cycles with 150 features	98.21	1.016
50 cycles with 10 features	80.72	1.017
50 cycles with 50 features	96.75	1.017
50 cycles with 100 features	98.21	1.018
50 cycles with 150 features	98.32	1.017
100 cycles with 10 features	80.49	1.016
100 cycles with 50 features	96.86	1.017
100 cycles with 100 features	98.21	1.016
100 cycles with 150 features	**98.43**	1.016
200 cycles with 10 features	80.96	1.018
200 cycles with 50 features	97.98	1.018
200 cycles with 100 features	98.21	1.017
200 cycles with 150 features	**98.43**	1.018
500 cycles with 10 features	80.85	1.019
500 cycles with 50 features	97.76	1.015
500 cycles with 100 features	98.21	1.016
500 cycles with 150 features	98.32	1.016
Lasso (Subspace)		
10 cycles with 100 features	96.60	57.1 s
10 cycles with 150 features	97.22	52.7 s
50 cycles with 100 features	96.28	58.1 s
50 cycles with 150 features	97.59	1.138
100 cycles with 100 features	96.27	1.057
100 cycles with 150 features	97.51	1.215
200 cycles with 100 features	95.52	1.867
200 cycles with 150 features	97.03	1.428
500 cycles with 100 features	96.45	2.469
500 cycles with 150 features	97.48	2.655

Note: Bold font indicates the best prediction accuracy.

4.2. Empirical data

There are two large datasets that we used in the empirical studies. The first is the Tezpur University Android Malware Dataset (TUANDROMD) from the UCI Machine Learning Repository where the binary response or target variable is malware vs goodware. This dataset contains 4465 observations and 241 features. The second dataset is the fraud detection bank data from Kaggle, which contains 20,468 observations and 113 features. Note that these two datasets were those with a large number of features we could find from an open-source. In the implementation of all models and methods using R, we again have a different combination of bootstrap cycles and features, which are shown in Tables 2–3.

Results from the TUANDROMD dataset in Table 2 where the test set accounts for 80% of all observations show that the random subspace logistic regression for 100 and 200 bootstrap cycles with 150 features returns the best prediction accuracy at 98.43% with the computational time of 1.016 and 1.018, respectively. The second best models and methods are the random subspace logistic for 10 cycles with 50 features and 50 cycles with 150 features and the neural network at 98.32%. In most cases, the computational time is not very different across models and methods except those of random subspace lasso logistic regression. Note that we chose to implement the random subspace lasso logistic with only 100 and 150 features, which are sufficiently large for the lasso because even with the lower number of features (50 features) the random subspace logistic still performs better than that of the lasso.

The results shown in Table 3 are from the fraud detection bank dataset that has a lot more observations but fewer features. In the implementation, the training set accounts for 70% of the whole observations. The best performing model is the SVM with the prediction accuracy of 99.83% followed by the random forest (99.76%) and the neural network (99.71%) while the computation time is still close to each other.

Table 3
Results from fraud detection bank dataset.

Model/Method	Accuracy (%)	Comp. time (mins)
Logit (All)	99.14	1.020
Lasso (All)	99.63	1.020
Random Forest	99.76	1.018
Neural Network	99.71	1.018
SVM	**99.83**	1.018
Logit (Subspace)		
10 cycles with 10 features	75.67	1.017
10 cycles with 50 features	99.44	1.017
10 cycles with 80 features	99.41	1.017
50 cycles with 10 features	81.53	1.017
50 cycles with 50 features	99.32	1.017
50 cycles with 80 features	99.46	1.016
100 cycles with 10 features	81.73	1.011
100 cycles with 50 features	99.39	1.017
100 cycles with 80 features	99.51	1.018
200 cycles with 10 features	78.92	1.017
200 cycles with 50 features	98.93	1.018
200 cycles with 80 features	99.46	1.018
500 cycles with 10 features	79.51	1.016
500 cycles with 50 features	99.21	1.017
500 cycles with 80 features	99.46	1.016
Lasso (Subspace)		
10 cycles with 80 features	96.45	54.7 s
50 cycles with 80 features	95.07	1.016
100 cycles with 80 features	95.35	1.072
200 cycles with 80 features	95.13	1.323
500 cycles with 80 features	95.23	1.957

Note: Bold font indicates the best prediction accuracy.

For the two random subspace models, the one from standard logistic regression still returns high prediction accuracy rates, especially for those with more bootstrap cycles and features, e.g., 99.51% from 100 cycles with 80 features and 99.46% from 50, 200, and 500 cycles with 80 features. This confirms our expectation that the random subspace logistic regression model can be a promising alternative for feature selection. For the random subspace lasso logistic, though it returns the worst performance among models and methods in this dataset, its prediction accuracy rates from various combinations are still above 95%.

Based on the results from both of simulation and empirical studies, we can conclude that our proposed random subspace logistic regression performs well and can be used as a wrapper method to select flat features for large datasets. The random subspace lasso logistic regression, as an embedded method, can also be a competing tool for the same purpose but we need to explore further to improve its performance. It is worth emphasizing here that different datasets have different characteristics and complexities, not a single model or method can always perform well for all datasets.

5. Concluding remarks

In the big data era, feature selection is a popular technique that has been used and developed regularly. Logistic regression is a wrapper method that is widely used for feature classification. Wrapper methods generally perform well in terms of classification or prediction but are computationally expensive. In this study, we apply the concept of random subspace regression to the standard and lasso logistic regression models where the lasso logistic is an embedded method. In this method, the features are randomly selected through bootstrap cycles. The results from simulation and empirical studies indicate that our proposed random subspace logistic regression models perform reasonably well. The

Table A.1
Comparison of classification from feature selection methods.

Author (Year)	Algorithm	Method	Data type	Classification method
Abe and Kudo (2006)	LDF, QDF, **kNN**, C4.5 SUB, NNC, SVM	Wrapper	Numerical	LSVM
Balamurugan and Rajaram (2009)	**Bayes' Theorem**, Wrapper Subset, Consistency subset, InfoGain, GainRatio, OneR, Chi-Square,	Filter	Binary and Numerical	SVM
Batra and Sabharwal (2014)	**MCM**, RelieF, FCBF	Filter	Binary and Numerical	SVM
Chang and Chen (2010)	**AMFES**, RFE, Correlation, Baseline	Embedded	Multiple	LSVM
Chang and Liu (2012)	**AMFES**, RFE, Correlation, SVM, Stepwise, NAMEFES	Embedded	Multiple	SVM
Duan et al. (2005)	**SVM-RFE**, MSVM-RFE	Wrapper	Numerical	SVM
Forman (2003)	**BNS**, Acc2, Information Gain	Wrapper	Categorical	SVM
Hall and Holmes (2003)	**Forward Selection**, CFS, RelieF	Wrapper	Numerical	Nav̈e Bayes
Hwang et al. (2017)	**RFE**, BE-VSSC **k-VSSC**, kNN, LR, Correlation, CART	Embedded & Wrapper	Numerical	SVM
Maldonado and Weber (2009)	RFE-SVM, FSV, **HO-SVM**, Fisher Score	Wrapper	Numerical	SVM
Mandal et al. (2021)	**MI**, **CS**, **XV**, **RelieF**	Hybrid	Multiple	KNN, SVM, Nav̈e Bayes
Su and Yang (2008)	ANN, **SVM-based L-J**	Wrapper	Numerical	SVM
Weston et al. (2000)	Pearson, **SVMs**, Fisher Score, Kolmogorov–Smirnov	Wrapper	Numerical and Image	SVM
Zhu and Yang (2013)	SFFS, SBFS, **AP-SFS**, AP-SBFS	Wrapper	Numerical	LDA, NBC, KNN

Note: Bold font indicates the best algorithm.

random subspace logistic regression can be a promising alternative for flat feature selection.

Given the favorable results from the proposed random subspace logistic regression, there are still three areas that can be explored further for future research. Firstly, in the aggregation of predicted values during the bagging process, alternative weighting schemes, e.g., other information or model selection criteria, should be considered to improve the prediction accuracy. Secondly, in alternative to the Newton–Raphson algorithm, other methods such as variational Bayes can be designed and used to estimate the random subspace logistic regression models to overcome the computational burden and improve the search for better parameters. Lastly, further investigation is needed for the random subspace lasso logistic regression. Though it is not the best performing model, its formulation is suitable for big data analysis and that can be improved.

CRediT authorship contribution statement

Nuttanan Wichitaksorn: Conceptualization, Methodology, Validation, Writing – original draft, Review and editing, Supervision. **Yingyue Kang:** Methodology, Programming, Software, Validation, Visualization, Writing – review. **Faqiang Zhang:** Methodology, Programming, Software, Validation, Visualization, Writing – review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data are already available from the open sources.

Acknowledgments

We would like to thank the Editor-in-Chief, the Associate Editor, and the reviewers for their feedback that greatly improves the manuscript.

Appendix

See Table A.1.

References

Abe, N., & Kudo, M. (2006). Non-parametric classifier-independent feature selection. *Pattern Recognition*, *39*(5), 737–746.

Arauzo-Azofra, A., Aznarte, J. L., & Benítez, J. M. (2011). Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, *38*, 8170–8177.

Ayalew, L., & Yamagishi, H. (2005). The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda–Yahiko Mountains, Central Japan. *Geomorphology*, *65*(1–2), 15–31.

Balamurugan, S. A. A., & Rajaram, R. (2009). Effective and efficient feature selection for large-scale data using Bayes' theorem. *International Journal of Automation and Computing*, *6*(1), 62–71.

Batra, S. S., & Sabharwal, S. (2014). Feature selection through minimization of the VC dimension. arXiv preprint. arXiv:1410.7372.

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *International Journal of Intelligent Systems in Accounting, Finance and Management*, *13*(3), 133–150.

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, *39*(227), 357–365.

Bertoni, A., Folgieri, R., & Valentini, G. (2005). Bio-molecular cancer prediction with random subspace ensembles of support vector machines. *Neurocomputing*, *63*, 535–539.

Bins, J., & Draper, B. A. (2001). Feature selection from huge feature sets. In *Proceedings of eighth IEEE international conference on computer vision* (pp. 159–165).

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). *Feature selection for high-dimensional data*. Cham: Springer International Publishing.

Boot, T., & Nibbering, D. (2019). Forecasting using random subspace methods. *Journal of Econometrics*, *209*, 391–406.

Brown, G., Pocock, A., Zhao, M. J., & Luján, M. (2012). Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, *13*(1), 27–66.

Chadwick, D., Arch, B., Wilder-Smith, A., & Paton, N. (2006). Distinguishing dengue fever from other infections on the basis of simple clinical and laboratory features: application of logistic regression analysis. *Journal of Clinical Virology*, *35*(2), 147–153.

Chang, F., & Chen, J. (2010). An adaptive multiple feature subset method for feature ranking and selection. In *Proceedings of international conference on technologies and applications of artificial intelligence* (pp. 255–262).

Chang, F., & Liu, C. C. (2012). *Ranking and selecting features using an adaptive multiple feature subset method*: *Technical report no. TR-IIS-12-005*, Institute of Information Science, Academia Sinica.

Chen, R.-H., Dewi, C., Huang, S.-W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(52), http://dx.doi.org/10.1186/s40537-020-00327-4.

Cheng, Q., Varshney, P. K., & Arora, M. K. (2006). Logistic regression for feature selection and soft classification of remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, *3*(4), 491–494.

Dash, M., & Liu, H. (2000). Feature selection for clustering. In *Proceedings of the 4th Pacific-Asia conference on knowledge discovery and data mining* (pp. 110–121).

Davis, L. (1991). Bit-climbing, representational bias, and test suit design. In *Proceedings of international conference on genetic algorithm* (pp. 18–23).

Duan, K. B., Rajapakse, J., Wang, H., & Azuaje, F. (2005). Multiple SVM-RFE for gene selection in cancer classification with expression data. *IEEE Transactions on Nanobioscience*, *4*(3), 228–234.

Eftekhar, B., Mohammad, K., Ardebili, H. E., Ghodsi, M., & Ketabchi, E. (2005). Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data. *BMC Medical Informatics and Decision Making*, *5*(1), 1–8.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, *5*(9), 1531–1555.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*, 1289–1305.

Gu, Q., Li, Z., & Han, J. (2011). Generalized fisher score for feature selection. In *Proceedings of the twenty-seventh conference on uncertainty in artificial intelligence* (pp. 266–273).

Guan, S. U., Liu, J., & Qi, Y. (2004). An incremental approach to contribution-based feature selection. *Journal of Intelligent Systems*, *13*(1), 15–42.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1), 389–422.

Hall, M. A. (2000). *Correlation-based feature selection of discrete and numeric class machine learning*: *Working paper 00/08*, Hamilton, New Zealand: University of Waikato, Department of Computer Science, https://hdl.handle.net/10289/1024.

Hall, M. A., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, *15*(6), 1437–1447.

Hart, P. E., Stork, D. G., & Duda, R. O. (2000). *Pattern classification*. Hoboken, New Jersey: Wiley.

He, X., Cai, D., & Niyogi, P. (2005). Laplacian score for feature selection. In *Proceedings of the eighteenth international conference on neural information processing systems* (pp. 507–514).

Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics*, http://dx.doi.org/10.1155/2015/198363.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Hsu, C. N., Huang, H. J., & Dietrich, S. (2002). The ANNIGMA-wrapper approach to fast feature selection for neural nets. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *32*(2), 207–212.

Huang, D., Cai, X., & Wang, C.-D. (2019). Unsupervised feature selection with multi-subspace randomization and collaboration. *Knowledge-Based Systems*, *182*, Article 104856.

Hwang, K., Kim, D., Lee, K., Lee, C., & Sungsoo, P. (2017). Embedded variable selection method using signomial classification. *Annals of Operations Research*, *254*(1), 89–109.

Isachenko, R. V., & Strijov, V. V. (2018). Quadratic programming optimization with feature selection for nonlinear models. *Lobachevskii Journal of Mathematics*, *39*(9), 1179–1187.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (p. 18). New York: Springer.

Khaire, U. M., & Dhanalakshmi, R. (2019). Stability of feature selection algorithm: A review. *Journal of King Saud University - Computer and Information Sciences*, http://dx.doi.org/10.1016/j.jksuci.2019.06.012.

Khandezamin, Z., Naderan, M., & Rashti, M. J. (2020). Detection and classification of breast cancer using logistic regression feature selection and GMDH classifier. *Journal of Biomedical Informatics*, *111*, Article 103591.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on machine learning* (pp. 249–256).

Kononenko, I., Šimec, E., & Robnik-Šikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, *7*(1), 39–55.

Lai, C., Reinders, M. J. T., & Kononenko, L. W. (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters*, *27*(10), 1067–1076.

Lee, S. A. R. O. (2005). Application of logistic regression model and its validation for landslide susceptibility mapping using GIS and remote sensing data. *International Journal of Remote Sensing*, *26*(7), 1477–1491.

Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *Speech and natural language: Proceedings of a workshop held at Harriman, New York*.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, *50*(6), 1–45.

Li, H., Lee, Y.-C., Zhou, Y.-C., & Sun, J. (2011). The random subspace binary logit (RSBL) model for bankruptcy prediction. *Knowledge-Based Systems*, *24*(8), 1380–1388.

Li, J., & Liu, H. (2017). Challenges of feature selection for big data analytics. *IEEE Intelligent Systems*, *32*(2), 9–15.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), Article 491502.

Ma, S., & Huang, J. (2008). Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, *9*(5), 392–403.

Maldonado, S., & Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, *179*(13), 2208–2217.

Mandal, M., Singh, P. K., Ijaz, M. F., Shafi, J., & Sarkar, R. A. (2021). Tri-stage wrapper-filter feature selection framework for disease classification. *Sensors*, *21*(16), 5571. http://dx.doi.org/10.3390/s21165571.

Mao, J., Mohiuddin, K., & Jain, A. K. (1994). Parsimonious network design and feature selection through node pruning. In *Proceedings of the twelfth international conference on pattern recognition* (pp. 622–624).

Meyer, P. E., Schretter, C., & Bontempi, G. (2008). Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, *2*(3), 261–274.

Miao, J., & Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, *91*, 919–926.

Michalak, K., & Kwasnicka, H. (2006). Correlation-based feature selection strategy in neural classification. In *Proceedings of the sixth international conference on intelligent systems design and applications* (pp. 741–746).

Ng, A. Y. (2004). Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning* (p. 78). ACM.

Ng, S. (2017). Opportunities and challenges: Lessons from analyzing terabytes of scanner data. In *Advances in economics and econometrics, eleventh world congress* (pp. 1–34).

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238.

Pham, B. T., Prakash, I., & Bui, D. T. (2018). Spatial prediction of landslides using a hybrid machine learning approach based on random subspace and classification and regression trees. *Geomorphology*, *303*, 256–270.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (2014). *C4.5: Programs for machine learning*. Elsevier.

Ren, Y.-Z., Zhang, G.-J., & Yu, G.-X. (2011). Random subspace based semi-supervised feature selection. In *Proceedings of the 2011 international conference on machine learning and cybernetics* (pp. 113–118).

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, *53*(1), 23–69.

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–2517.

Singh, S., Kubica, J., Larsen, S., & Sorokina, D. (2009). Parallel large scale feature selection for logistic regression. In *Proceedings of the 2009 SIAM international conference on data mining* (pp. 1172–1183).

Su, C. T., & Yang, C. H. (2008). Feature selection for the SVM: An application to hypertension diagnosis. *Expert Systems with Applications*, *34*(1), 754–763.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. In *Data classification: Algorithms and applications* (pp. 37–64). CRC Press.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, *28*(2), 3–27.

Wang, G., & Ma, J. (2011). Study of corporate credit risk prediction based on integrating boosting and random subspace. *Expert Systems with Applications*, *38*(11), 13871–13878.

Weber, G., Vinterbo, S., & Ohno-Machado, L. (2004). Multivariate selection of genetic markers in diagnostic classification. *Artificial Intelligence in Medicine*, *31*(2), 155–167.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. In *Advances in neural information processing systems, vol. 13*.

Yang, H., & Moody, J. (1999). Data visualization and feature selection: New algorithms for nongaussian data. In *Proceedings of the twelfth international conference on neural information processing systems* (pp. 687–693).

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the twentieth international conference on machine learning* (pp. 856–863).

Zhu, K., & Yang, J. (2013). A cluster-based sequential feature selection algorithm. In *Proceedings of the ninth international conference on natural computation* (pp. 848–852).

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