South African Statistical Journal
Vol. 57, No. 2, pp. 89–129
https://doi.org/10.37920/sasj.2023.57.2.2

An automated exact solution framework towards solving the logistic regression best subset selection problem

Thomas K. van Niekerk[1], *Jacques V. Venter*[2] *and Stephanus E. Terblanche*[1]

[1]School of Industrial Engineering, North-West University, Potchefstroom, South Africa
[2]Centre for Business Mathematics and Informatics, North-West University, Potchefstroom, South Africa

An automated logistic regression solution framework (ALRSF) is proposed to solve a mixed integer programming (MIP) formulation of the well known logistic regression best subset selection problem. The solution framework firstly determines the optimal number of independent variables that should be included in the model using an automated cardinality parameter selection procedure. The cardinality parameter dictates the size of the subset of variables and can be problem-specific. A novel regression parameter fixing heuristic that utilises a Benders decomposition algorithm is applied to prune the solution search space such that the optimal regression parameter values are found faster. An optimality gap is subsequently calculated to quantify the quality of the final regression model by considering the distance between the best possible log-likelihood value and a log-likelihood value that is calculated using the current parameter values. Attempts are then made to reduce the optimality gap by adjusting regression parameter values. The ALRSF serves as a holistic variable selection framework that enables the user to consider larger datasets when solving the best subset selection logistic regression problem by significantly reducing the memory requirements associated with its mixed integer programming formulation. Furthermore, the automated framework requires minimal user intervention during model training and hyperparameter tuning. Improvements in quality of the final model (when considering both the optimality gap and computing resources required to achieve a result) are observed when the ALRSF is applied to well-known real-world UCI machine learning datasets.

Keywords: Best subset selection, Independent variable selection, Logistic regression, Mixed integer programming.

1. Introduction

Binary classification problems play an important role within the broader field of statistical modelling and can be solved by using algorithms such as support vector machines, neural networks, decision tree based methods, k-nearest-neighbours and logistic regression (Civitelli et al., 2021). Logistic regression is by far one of the most popular approaches which is supported by a wealth of literature and research (Civitelli et al., 2021). In general, logistic regression algorithms are straightforward to

Corresponding author: Thomas K. van Niekerk (thomasvn67@gmail.com)
MSC2020 subject classifications: 90-XX, 62-XX

implement, can be applied to relatively large machine learning problems and provide a measure of interpretability (Civitelli et al., 2021; Stoltzfus, 2011).

An important area of research (with reference to solving binary classification problems) is finding the optimal subset of independent variables from a set of independent variables $j \in p = \{1, 2, ..., |p|\}$ and observations $i \in n = \{1, 2, ..., |n|\}$ to include in the final model during the training phase. Let X_i denote a $p \times 1$ input vector of independent variable values for the ith observation in a dataset with n records, where the jth entry of X_i correspond to the jth variable. Furthermore, let $Y_i \in \{0, 1\}$ be a binary dependent or target variable that follows a Bernoulli distribution where $P(Y_i = 1|X_i) = (1 + \exp(-\beta^T X_i))^{-1}$. Within a regression model setting, the independent variable entries in X_i represent a vector of p predictors that are used as inputs in a machine learning model in an attempt to predict the dependent variable Y_i. It is assumed that each observation (X_i, Y_i) is independently sampled from the same population. The selected subset of input variables should accurately capture the nature of the problem instance while still allowing for adequate generalisation performance. Best subset selection (in its most fundamental form) describes a process that attempts to select the most relevant and statistically significant set of independent variables by optimising some objective function $f(\beta; \mathbf{X}, \mathbf{Y})$ with respect to a vector of regression parameters $\beta \in \mathbb{R}^p$, where β_j denotes the jth regression parameter associated with the jth input variable (with β_0 being the intercept term). Mathematically, the problem can be defined as

$$\max_{\beta} f(\beta; \mathbf{X}, \mathbf{Y}), \tag{1}$$

or

$$\min_{\beta} f(\beta; \mathbf{X}, \mathbf{Y}), \tag{2}$$

subject to

$$||\beta||_0 \leq c, \tag{3}$$

where $||.||_0$ is the L_0-norm, $||\beta||_0 = \sum_{j \in p} I(\beta_j > 0)$ and (3) is known as a cardinality selection constraint that allows the regression model to contain at most c of the p available independent variables (Zhang et al., 2018; Venter, 2020). The composition of $f(\beta; \mathbf{X}, \mathbf{Y})$ depends on the type of regression algorithm that is applied. In the case of logistic regression, the log-likelihood function is typically used as the objective function $f(\beta; \mathbf{X}, \mathbf{Y})$. Since the log-likelihood function is concave, the objective function in (1) can be modified according to

$$\max_{\beta} f(\beta; \mathbf{X}, \mathbf{Y}) = \log(L(Y_1, ..., Y_n)) = \sum_{i \in n} Y_i(\beta^T X_i) - \sum_{i \in n} \log(1 + \exp(\beta^T X_i)), \tag{4}$$

where $\beta^T X_i$ is equal to $\sum_{j \in p} \beta_j X_{ij}$. The choice between (1) or (2) depends on the objective function being used. As an alternative to the log-likelihood function, the modeller may employ the logistic loss function, which is defined as

$$\min_{\beta} f(\beta; \mathbf{X}, \mathbf{Y}) = \sum_{i \in n} \log(1 + \exp(-Y_i(\beta^T X_i)). \tag{5}$$

Since the log-likelihood is concave, the negative of the log-likelihood (logistic loss function) will

be convex. This explains why maximisation of (4) and minimisation of (5) will yield identical results. Different formulations of the best subset selection problem make use of either the log-likelihood or the logistic loss objective function.

It is well known that training a logistic regression classifier on too many independent variables may result in the model overfitting on the training set, which in turn degrades the predictive performance of the final selected model. This is especially true when the number of independent variables p greatly exceeds the number of training observations n such that $p > n$ (Naganuma et al., 2021). Numerous studies have attempted to solve (to optimality) the logistic regression best subset selection problem by using a variation of three main MIP logistic regression formulations that are commonly found in the literature. The first approach attempts to find the optimal subset of variables by minimising a logistic loss objective function that contains a regularisation term, such as the L_0-norm, L_1-norm (lasso) or L_2-norm (ridge regression) (Dedieu et al., 2021; Insolia et al., 2021; Kamiya et al., 2019; Takano and Miyashiro, 2020). A second approach involves a piecewise linear approximation of the non-linear logistic loss function, where the objective function is adjusted such that the best model is chosen based on a problem-specific information criterion, like the Akaike information criterion (AIC) or Bayesian information criterion (BIC) (Sato et al., 2016; Civitelli et al., 2021). Finally, we consider an alternative approach proposed by Venter (2020), who, similar to Sato et al. (2016), also makes use of a piecewise linear approximation of the logistic regression objective function. The difference, however, is that the use of an approximated log-likelihood by Venter (2020) leads to an underestimation of the optimal log-likelihood, while an overestimation is observed because of the logistic loss used by Sato et al. (2016). Furthermore, a strict and explicit cardinality constraint (instead of some information criterion) limits the number of variables included in the final model(s) produced by Venter (2020). These overarching formulations and the relevant underlying theory will be discussed in Section 2. A detailed account of the limitations associated with each approach will be provided to further highlight the contributions of the proposed automated logistic regression solution framework presented herein.

One of the main drawbacks of deterministic MIP solution frameworks relates to the computational burden and resource constraints imposed by these formulations, which in turn limits the size and dimensionality of the prediction problem that can be considered. According to Miller (2002), best subset selection MIP approaches are exponential-time procedures, which implies that an exponential increase in execution time is observed alongside an increase in the number of independent variables. In fact, best subset selection problems are NP-hard by nature (Natarajan, 1995; Venter, 2020). Some authors note that attempts to perform best subset selection procedures using commercially available statistical software become intractable when the input dataset contains approximately 40 - 75 independent variables (Lund, 2017).

The framework proposed in this paper mainly builds on the logistic regression MIP formulation presented by Venter (2020) because of the favourable statistical characteristics that it exhibits. With the above in mind, we set out to develop a novel automated solution framework that can be used to solve large-scale best subset selection logistic regression problems that are formulated as MIPs. The framework in question mainly builds on the formulation proposed by Venter (2020) because of its favourable statistical attributes. Additionally, substantial attempts are made to address some of the limitations inherent to logistic regression MIP problems which are commonly found when dealing with best subset selection. Firstly, the automated solution framework (dubbed the ALRSF) produces

interpretable and parsimonious models that generalise well on unseen data by avoiding the use of regularisation terms and biased regression parameters. This is done by incorporating an explicit cardinality constraint into the formulation. Secondly, the ALRSF requires minimal user intervention (unlike some of the other formulations found in literature) by automatically determining the most appropriate cardinality parameter and iteratively adjusting the parameter search space during model training. As such, the regression parameters of the model rely more on the underlying patterns that are present in the data and less on input from the model developer. Finally, and perhaps most importantly, the ALRSF exhibits significant improvements in computational memory requirements when presented with best subset selection problems. These improvements enable the user to consider larger datasets that would otherwise be intractable when solved with the MIP approaches mentioned earlier.

The remainder of this paper can be subdivided into six main sections. Section 2 provides a background on best subset selection and related work, accompanied by appropriate theoretical considerations that underpin the proposed ALRSF. The contribution of the ALRSF to existing literature is also presented. Sections 3 and 4 introduce the reader to the proposed best subset selection MIP formulation with relevant modifications that allow it to be used within an automated logistic regression setting. The logic of the ALRSF is then summarised in Section 4 followed by model verification in Section 5. Empirical results are presented in Section 6 where predictive models are applied to three real-world datasets in order to evaluate the performance of the proposed ALRSF and to compare it against existing best subset MIP approaches. Section 6 further highlights the contribution of the ALRSF to the intersecting fields of variable selection and computational efficiency when dealing with sizeable datasets. Concluding remarks and recommendations with respect to future research are discussed in Section 7.

2. Background and rationale

In this section, we elaborate on the concept of best subset selection within a MIP problem setting where regularisation terms, information criteria or explicit cardinality parameters are incorporated into the model objective function to select the optimal subset of variables from the set of p predictors. Firstly, MIP formulations that rely on regularisation techniques will be considered. In its most simplistic form, regularisation MIP formulations can be defined as[1]

$$\min_{\beta} \quad \frac{1}{n} f(\beta; \mathbf{X}, \mathbf{Y}) + \lambda ||\beta||_{q_2}^{q_1}, \tag{6}$$

where $||.||_{q_2}^{q_1}$ controls the type of regularisation used for subset selection. When $q_1 = q_2 = 1$, the popular lasso regression approach is followed. Alternatively, setting $q_1 = q_2 = 2$ forces the model to apply a ridge regression penalty. Unlike lasso and ridge regression (which are not NP-hard), we can specify a strict penalty by setting $q_1 = 1$ and $q_2 = 0$, which will use the L_0-norm and perform best subset selection. Incorporating a combination of the aforementioned regularisation terms into a single objective function can also be considered in an attempt to overcome the limitations associated

[1] Note that the objective function is minimised when the logistic loss is used. Alternatively, the objective function will be maximised when the log-likelihood is employed, with the "+" sign in front of the regularisation term (the second term of the objective function) changed to a "-" sign.

with each individual approach (see Dedieu et al. (2021) for more details). The regularisation-based MIP formulation proposed by Dedieu et al. (2021) utilises L_0-L_q-norms as penalty terms and is shown in (7), where $\lambda_0 \geq 0$ controls the number of non-zero regression parameters and enforces model sparsity, while $\lambda_q \geq 0$ controls the shrinkage of model parameters.

$$\min_{\beta} \quad \frac{1}{n} f(\beta; \mathbf{X}, \mathbf{Y}) + \lambda_0 ||\beta||_0 + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2. \tag{7}$$

Each regularisation term can be included/excluded based on the value(s) selected for q_1, q_2, λ_0, λ_1 and λ_2. Dedieu et al. (2021) reports that the function $f(\beta; \mathbf{X}, \mathbf{Y})$ is interchangeable between the logistic loss function (defined in Section 1) or the hinge loss function. A detailed overview of each type of loss function can be found in the author's work. From Equation (7) we can group the logistic loss function, lasso and ridge regression terms into a single term that can be written as

$$h(\beta; \mathbf{X}, \mathbf{Y}, \lambda_1, \lambda_2) = \frac{1}{n} f(\beta; \mathbf{X}, \mathbf{Y}) + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2. \tag{8}$$

Substituting (8) into (7) leads to

$$\min_{\beta, z} \quad h(\beta; \mathbf{X}, \mathbf{Y}, \lambda_1, \lambda_2) + \lambda_0 \sum_{j \in p} z_j, \tag{9}$$

subject to

$$|\beta_j| \leq M z_j \quad \forall j \in p, \tag{10}$$

$$z_j \in \{0, 1\} \quad \forall j \in p. \tag{11}$$

The formulation in (9)–(11) is a special case of the regularised regression model presented in (6) and allows for variable selection by optimising some inherent cardinality parameter based on the z_j decision variable. When $z_j = 0$, the jth independent variable is excluded from the final subset by setting $\beta_j = 0$. Alternatively, $|\beta_j|$ is allowed to take on any value in the range $[0, M]$ when $z_j = 1$. The constant M in constraint (10) governs the range of values that the regression parameters are allowed to take on and should be sufficiently large (ideally, in a perfect world where we have unlimited computational power M should tend to infinity). Given that the L_0-norm describes the number of non-zero regression parameters (the number of parameters where $\beta_j \neq 0$), we can replace $\lambda_0 ||\beta_0||$ with $\lambda_0 \sum_{j \in p} z_j$ to perform the same task. A tighter formulation of the optimisation problem in (8)–(11) was proposed by Dedieu et al. (2021) and is summarised in (12)–(17):

$$\min_{\beta, z, \omega} \quad h(\omega; \mathbf{X}, \mathbf{Y}, \lambda_1, \lambda_2) + \lambda_0 \sum_{j \in p} z_j, \tag{12}$$

subject to

$$|\beta_j| \leq M z_j, \quad j \in A', \tag{13}$$

$$\omega = \beta_j^t - \sum_{j \in A} e_j \beta_j^t (1 - z_j) + \sum_{j \in A'} e_j \beta_j, \tag{14}$$

$$\sum_{j \in A} z_j \geq |A| - m, \tag{15}$$

$$\sum_{j \in A'} z_j \leq m, \tag{16}$$

$$z_j \in \{0, 1\} \quad \forall j \in p. \tag{17}$$

Dedieu et al. (2021) makes use of a two-step iterative local search algorithm to first prune the solution search space. The best solution discovered as part of the local search algorithm is then used as an input to the tightened MIP formulation presented in (12)–(17) in an attempt to either improve the best integer solution or provide an optimality guarantee. The first algorithm used (as part of the initial two-step local search algorithm) is a coordinate descent approach followed by a local combinatorial search algorithm. The reader is referred to Dedieu et al. (2021) for a detailed explanation of this iterative algorithmic approach. Finally, the tightened MIP formulation can be used to try and improve on the local solutions discovered while also attempting to provide an optimality guarantee. Let $A \subset p$ denote a subset of variables from all available independent variables p with the complement of A denoted as A'. Then the support set can be defined as $A = \text{Supp}(\beta)$ where $\beta \neq 0$. Let $A^1 \subset A$ and $A^2 \subset A'$ denote two sets with size of at most m (the value of m will be discussed shortly). Let β^t denote the vector of regression parameters that were generated at iteration t of the local search algorithm (the combination of the coordinate descent and combinatorial search approaches)[2]. As part of the combinatorial search step a $p \times p$ matrix B^A is defined, where the jth row of B^A is equal to $\sum_{j \in p} e_j$ if j element of A and zero otherwise. This means that for any $\beta \in \mathbb{R}^p$, $(B^A\beta)_j = \beta_j$ when $j \in A$ and $(B^A\beta)_j = 0$ when $j \notin A$. This swap-and-solve logic is captured by the constraints in (14)–(17) and is facilitated by the binary decision variables $z_j, j = 1, ..., p$. Particularly when $z_j = 0 \ \forall j \in A$ it means that the jth variable in A^1 should be removed from support set A. Alternatively, when $z_j = 1 \ \forall j \in A'$ it means that the jth variable in A^2 should be added to the support set A. The size of m controls the number of variables that can be considered for inclusion into the support A. A large value for m will generally result in extended model execution times.

Dedieu et al. (2021) go on to develop another MIP formulation which is solved using an integrability generation algorithm. Let A represent the support and A' the complement of A. Once again the heuristic-based solution algorithm (which is a combination of coordinate descent with local combinatorial search) is used to prune the solution search space. The support A is then initialised with the variables identified by the initial heuristic search algorithm. The proposed MIP formulation can be defined as

$$\min_{\beta, z} \quad h(\beta; \mathbf{X}, \mathbf{Y}, \lambda_1, \lambda_2) + \lambda_0 \sum_{j \in p} z_j, \tag{18}$$

subject to

$$|\beta_j| \leq M z_j, \quad \forall j \in p, \tag{19}$$

$$z_j \in [0, 1], \quad \forall j \in A', \tag{20}$$

$$z_j \in \{0, 1\}, \quad \forall j \in A, \tag{21}$$

[2]For the remainder of this paper, the reader should note that an uppercase superscript T (e.g. D^T) refers to the transpose of a vector/matrix, whereas a lowercase superscript t (e.g. β^t) refers to the tth iteration.

where $z_j \in A'$ in (20) is allowed to be continuous and take on any value between zero and one, while integrality constraints are placed on $z_j \in A$ in (21). Allow β^t and z^t to denote the regression parameters and binary selection variables at iteration t. At iteration $t+1$ update $A = A \cup \{j | z_j^t \neq 0, j \in A'\}$ and then solve the MIP in (18)–(21) with warm-starting functionality of the MIP solver activated. This means that the solution at iteration t is used as starting point at iteration $t+1$. Since the MIP is a relaxation of (9)–(11), it provides a lower bound to the original problem i.e. logistic loss that is produced by the regression model (9)–(11) will be greater than or equal to the logistic loss obtained when using (18)–(21). As such, at each iteration t, the MIP in (18)–(21) aims to improve the MIP lower bound while also allowing an optimality gap to be calculated. The solution algorithm proposed by Dedieu et al. (2021) is capable of solving high-dimensional problems with a variable set of up to 50 000 independent variables as a result of the novel integrability generation logic applied. In general, the empirical results presented by the authors focused on solving sparse classification problems where $n \leq 1000$.

The lasso or L_1-norm regularisation has multiple attractive properties, some of which include its computational efficiency and scalability in the case of large-scale problem instances (Bertsimas et al., 2016; Hastie et al., 2020). One of its most useful characteristics is its ability to act as a true independent variable selection algorithm by setting some regression parameter estimates exactly equal to zero. Lasso regression does, however, introduce model bias by shrinking parameter estimates towards zero. This is especially true when independent variables in the design matrix have different measurement scales, where the betas associated with variables that have large true underlying regression parameters experience a greater degree of shrinkage (and, therefore, bias) (Bertsimas et al., 2016). In turn, the resulting shrinkage can adversely impact statistical inferences that are carried out when assessing the model, such as the evaluation of log-odds ratios (Heinze et al., 2018). Ridge regression suffers from the same drawbacks as lasso, but has the added disadvantage of producing larger models that contain many independent variables. This is because ridge regression only shrinks the regression parameters without setting them exactly equal to zero (Venter, 2020). As such, ridge regression should not be considered a true subset selection technique. Parameter bias as a result of regularisation can, however, be alleviated by performing a second round of model fitting, where a logistic regression model that only uses the variables identified by lasso is considered for inclusion. During this second round, no variable selection is performed (i.e. no regularisation terms are present in the model) and the goal is simply to generate improved parameter estimates with less bias for the purpose of statistical inference.

Regularisation techniques usually employ a penalisation parameter λ to control the magnitude of regularisation. The choice of this penalisation parameter (when manually selected) is subjective in nature and leads to a trade-off between variance and bias (Venter, 2020). The subjective nature of manually selecting the penalisation parameter can, however, be mitigated by an automated approach that uses a combination of cross-validation and grid search. When such an approach is utilised, a predefined grid of penalisation parameters is considered during various iterations of model fitting, where the most appropriate parameter value that corresponds to the best model performance is selected. Similarly, Bertsimas et al. (2016) proposes a heuristic-based approach to generate a regularisation path for the penalty parameters. It should be noted, however, that cross-validation and the approach proposed by Bertsimas et al. (2016) are heuristics by nature and as such can not provide an optimality guarantee with respect to the quality of the final model. This is because the complete

solution search space over the full range of penelisation parameter values cannot be considered simultaneously, since the optimality gap yielded by (6), (9) or (12) is specific to some constant value of λ. The optimality gap (which is used to provide an optimality guarantee) can be defined according to

$$\text{optimality gap} = \frac{bb - bi}{|bi| + \alpha},$$

where bb denotes the best bound of the MIP obtained and bi the best integer solution. An arbitrary small value α is added to the denominator to prevent the gap from being undefined when $|bi|$ is zero. The best bound in a maximisation problem can be interpreted as the bound obtained by a linear programming (LP) relaxation of the original problem. This means that all integer or binary variables are relaxed and allowed to take on any real value, which reduces the number of constraints that the problem needs to adhere to. The best integer solution is obtained by means of a branch-and-bound algorithm. The branch-and-bound algorithm is designed to solve combinatorial optimisation problems and a detailed overview of the algorithm is available in Section 4.2.2 of Venter (2020).

An alternative to variable selection using regularisation terms is rooted in the concept of information theory. Some of the most well-known information criteria include AIC and BIC. An IC-based subset selection problem that utilises the AIC solves

$$\min_{\beta,z} \quad 2f(\beta; \mathbf{X}, \mathbf{Y}) + 2(\sum_{j \in p} z_j + 1), \tag{22}$$

whereas a model that utilises the BIC is fitted by solving

$$\min_{\beta,z} \quad 2f(\beta; \mathbf{X}, \mathbf{Y}) + \log(|n|)(\sum_{j \in p} z_j + 1). \tag{23}$$

Using (22) and (23), Sato et al. (2016) proposes a logistic regression model that minimises a piecewise linear approximation of the logistic loss function while simultaneously performing subset selection via the inclusion of the AIC or BIC in the linearised objective function. Let γ_k denote a set of symmetric grid values, where $k \in K = \{1, 2, ..., |K|\}$. The authors approximate the non-linear logistic loss function in (5) by creating a series of linear functions that track the logistic loss curve from below and cut away non-feasible regions of the solution space. This relates to the pointwise maximisation of a set of linear inequalities which can be written as

$$\max\{f'(\gamma_k)(Y_i(X_i^T\beta) + f(\gamma_k)\} \quad \forall k = 1, ..., K. \tag{24}$$

These linear inequalities can be reformulated as a minimisation problem according to

$$\min\{\varphi | \varphi \geq f'(\gamma_k)(Y_i(X_i^T\beta) + f(\gamma_k)\} \quad \forall k = 1, ..., K. \tag{25}$$

The methodology proposed by Sato et al. (2016) is summarised in Figure 1 with the dashed lines representing the piecewise linear approximations of the logistic loss function. Since the approximation is below the logistic loss function, it can be described as an overestimation of the log-likelihood function, since the negative of the logistic loss yields the log-likelihood. The number

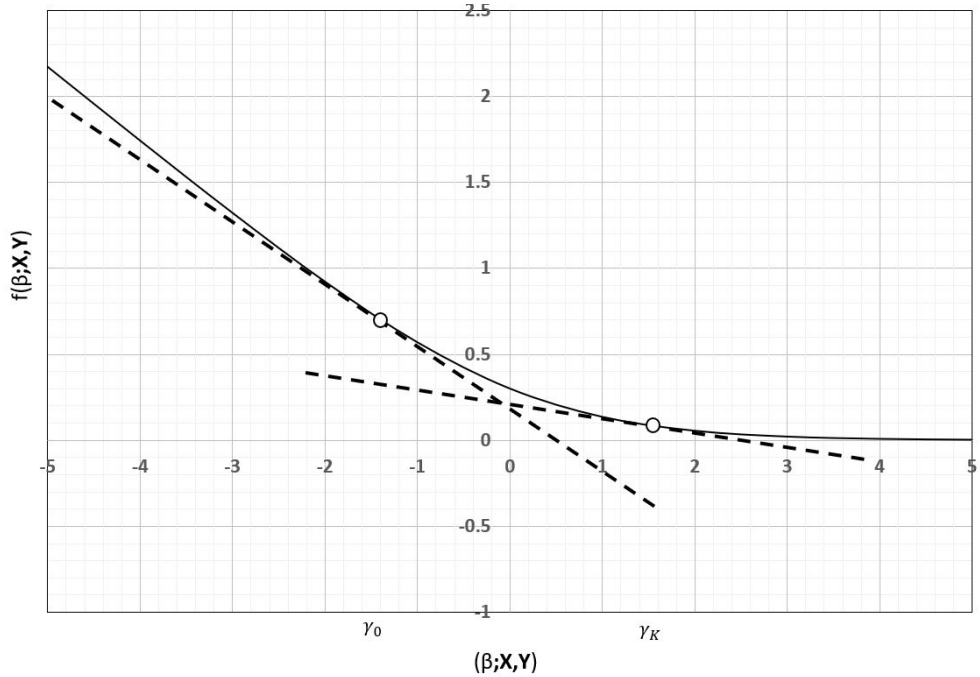

Figure 1. Piecewise linear overestimation of the logistic loss function

of tangent lines is defined by K (in Figure 1, $K = 2$ tangent lines are used). Larger numbers of linear splines result in an approximation that is closer to the non-linear logistic loss function.

The proposed piecewise linear approximation approach by Sato et al. (2016) can formally be defined as

$$\min_{\beta,z,\varphi} \quad 2\sum_{i\in n}\varphi_i + I(\sum_{j\in p} z_j + 1). \tag{26}$$

subject to

$$\varphi_i \geq f'(\gamma_k)(Y_i(X_i^T\beta) - \gamma_k) + f(\gamma_k) \quad \forall k \in K; \forall i \in n, \tag{27}$$

$$z_j = 0 \rightarrow \beta_j = 0 \quad \forall j \in p, \tag{28}$$

$$z_j \in \{0, 1\} \quad \forall j \in p, \tag{29}$$

where (27) represents a piecewise linear estimation of the logistic loss function as discussed in detail by Venter (2020) and Sato et al. (2016). Note that I in (26) does not denote an indicator function, but instead represents the information criterion that is applied during model fitting. When $I = 2$, the objective function in (22) is found and the AIC is used. Alternatively, when $I = \log(\cdot)$, the objective function (23) is used, which implies that the model is optimised with respect to the BIC.[3] The number of tangent lines K determines the accuracy of the linear approximation and, therefore, a trade-off exists between approximation accuracy and execution time (Sato et al., 2016). Variable

[3]It is important to note that I, used as part of the MIP formulation presented in (26)–(29), is a constant that facilitates switching between the use of either the Akaike or Bayesian information criterion. It does not represent an indicator function.

subset selection is facilitated by the selection decision variable z_j by means of (28) which is referred to as an indicator constraint or SOS1 (special ordered set), which implies that only one element in the set $\{1 - z_j, \beta_j\} \forall j \in p$ can be non-zero. This means that β_j should be set equal to exactly zero when $z_j = 0$ and non-zero otherwise. Constraint (28) is an alternative to a big-M formulation which can equivalently be defined as $\sum_{j \in p} |\beta_j| \leq M z_j$. It is important to note, however, that empirical evidence suggests that variable subset selection by means of minimising some information criterion (e.g. AIC or BIC) tends to favour larger models that are less parsimonious (Venter, 2020; Kutner et al., 2005).

Finally, we consider the piecewise linear approximation formulation for the logistic regression best subset selection problem that is proposed by Venter (2020). This formulation differs from that of Sato et al. (2016) in two noticeable areas, namely[4]

1. an explicit cardinality constraint is used instead of some information criterion to facilitate variable selection; and

2. instead of minimising the logistic loss (which results in an overestimation of the log-likelihood), a linear approximation of the log-likelihood is maximised (which results in an underestimation of the log-likelihood).

By critically evaluating the log-likelihood function defined in (4) it is clear that the first term is already linear. As such, only the second term $\sum_{i \in n} \log(1 + \exp(\beta^T X_i))$ needs to be linearised. The non-linear section is linearised by using $\Lambda_{ik} \in \mathbb{R}^{\{0,1\}}$ as an interval selection variable which relates the linearised grid point splines γ_k to the continuous estimates of $\beta^T X_i$. In other words, the decision variable Λ_{ik} identifies the appropriate line segment between two splines $[\gamma_k, \gamma_{k+1}]$ (used to approximate the log-likelihood function) that corresponds to the log-likelihood function value $f(\beta; \mathbf{X}, \mathbf{Y})$. The non-linear term of the log-likelihood function is therefore estimated by K linear line segments that connect at grid points γ_k, where $\min(\gamma_k) = -\Gamma$ and $\max(\gamma_k) = \Gamma$. The decision variable Λ_{ik} can alternatively be thought of as an interpolation variable. Figure 2 provides a visual representation of the piecewise linear underestimation of the log-likelihood function, where the dashed lines represent the piecewise linear approximation proposed by Venter (2020) and the solid black line denotes the non-linear log-likelihood.

The piecewise linear approximation of the log-likelihood attempts to fit a logistic regression model while simultaneously performing best subset selection by solving

$$\max_{\beta, \Lambda, z} \sum_{i \in n} \sum_{k \in K} (\gamma_k Y_i - \log[1 + \exp(\gamma_k)]) \Lambda_{ik}, \tag{30}$$

subject to

$$\sum_{k \in K} \Lambda_{ik} \gamma_k = \beta^T X_i \quad \forall i \in n, \tag{31}$$

[4] From Figure 1, observe that the optimal solution is found where the two lines intersect. This means that the linear approximation of the logistic loss function underestimates the non-linear function (the solid line in Figure 1). However, given that $-f(\beta; \mathbf{X}, \mathbf{Y})$ is the logistic loss function and $f(\beta; \mathbf{X}, \mathbf{Y})$ the log-likelihood, we note that an underestimation of the logistic loss function results in an overestimation of the log-likelihood. The opposite is true for the formulation proposed by Venter (2020).

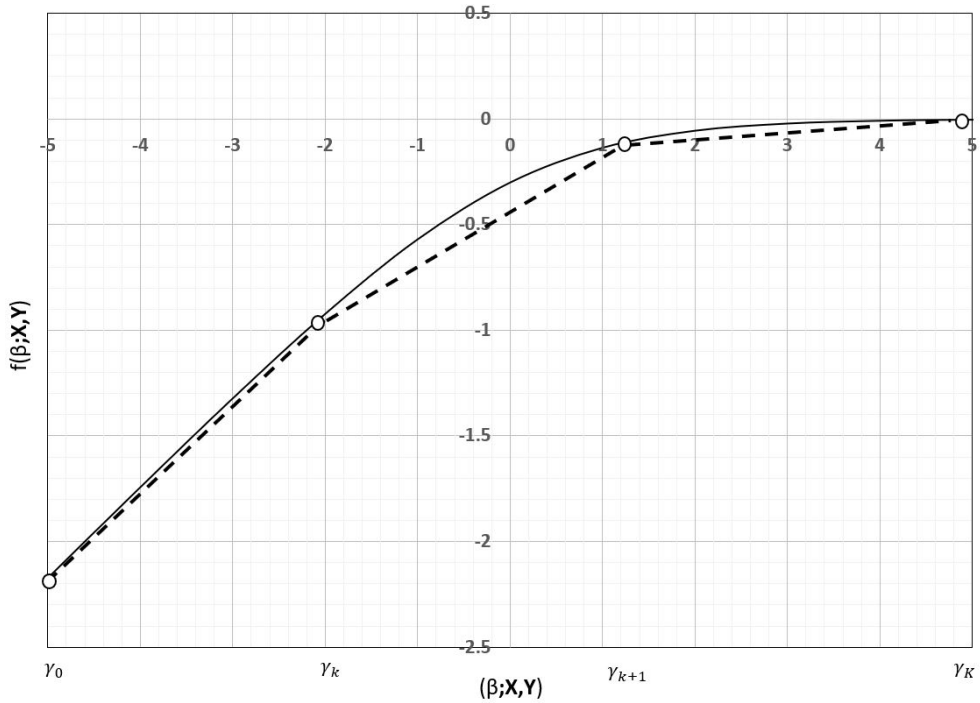

Figure 2. Piecewise linear underestimation of the log-likelihood function

$$\sum_{k \in K} \Lambda_{ik} = 1 \quad \forall i \in n, \tag{32}$$

$$0 \leq \Lambda_{ik} \leq 1 \quad \forall k \in K; \forall i \in n, \tag{33}$$

$$|\beta_j| \leq M z_j \quad \forall j \in p, \tag{34}$$

$$\sum_{j \in p} z_j \leq c. \tag{35}$$

Constraint (32) guides the selection of the appropriate linearised line segments in conjunction with (31) and (33) such that the log-likelihood is maximised. Constraints (34) and (35) are jointly referred to as cardinality selection constraints such that (35) ensures that a maximum of c independent variables are included in the model, while (34) allows the regression parameter $\beta_j \ \forall j \in p$ to default to 0 when $z_j = 0$ and range between $[-M, M]$ when $z_j = 1$. By using explicit cardinality constraints, the shrinkage effect (and, therefore the bias) introduced by regularisation is no longer present and the computational burden associated with the determination of the penalty term (e.g. cross-validation) is eliminated. In contrast to regularisation approaches, the MIP formulation proposed by Venter (2020) does not influence the interpretability of the regression parameters (Heinze et al., 2018). However, in all fairness, the proposed MIP formulation does require the specification of a cardinality value c that is used to limit the number of independent variables that can enter the model. In contrast to regularisation approaches, all possible variable combinations are considered for a specific cardinality value c, which means that the best subset MIP formulation can produce an optimality

gap and, therefore, a true measure of optimality (Hastie et al., 2020). This means that changing the value of c will only influence the number of independent variables included in the final subset and the final model performance, but (theoretically) for any given c, the best possible model is always obtained (the same cannot be said for regularised approaches or approaches that make use of p-values). Furthermore, an added benefit of the best subset MIP formulation is the possibility of explicitly controlling the sparsity of the model through c. Alternatively, no clear relationship exists between c and the penalty term λ that is used in regularisation approaches (Bertsimas et al., 2016).

Unfortunately, the approach in (30)–(35) requires some manual intervention for successful execution. The end-points of the grid $[-\Gamma, \Gamma]$ that describe the linearised approximation of the log-likelihood need to be manually calculated. This can be achieved by solving an LP relaxation of the original MIP problem formulation proposed by Venter (2020) that excludes all cardinality constraints. The linearised piecewise extreme points are then derived from the initial solution in a way which ensures that the complete search space is considered. According to the author, this type of manual intervention is undesirable if the objective is to automate the process of model fitting such that the optimal search space is determined by the algorithm and not the user. A detailed approach for finding the extreme points of the grid used to linearise the log-likelihood can be found in Venter (2020).

For the remainder of this paper, we use the formulation proposed by Venter (2020) to design our automated best subset logistic regression framework (our ALRSF). We address some of the drawbacks (as mentioned previously) associated with the model in (30)–(35) while attempting to automate the model fitting process and reduce the computational burden of best subset selection as much as possible.

3. Proposed MIP formulation

The proposed ALSRF combines the concept of a regression parameter fixing heuristic and Benders decomposition to allow for an improved deterministic solution framework that solves the best subset selection MIP problem within acceptable time frames while providing a guarantee of optimality. The solution framework is made up of three distinct steps which include the cardinality parameter selection step (automatically determining the most appropriate number of variables to include in the model), search space pruning step (automatically reducing the MIP search space that contains the range of possible values which regression parameters can take on in order to reduce run times) and an optimality gap calculation step (in order to provide a guarantee on the quality and optimality of the final regression model). This section presents the reader with the modifications made to the MIP formulation in (30)–(35), followed by a detailed exposition of the proposed novel ALRSF in Section 4.

First, let p' denote a subset of p such that $p' \subseteq p$. The subset p' can be defined as a set of independent variables for which the variance inflation factor (VIF) is larger than some threshold L. The subset p' therefore contains the indices of all the variables in the original set p that need to be excluded by the ALRSF based on the VIF criteria. Incorporating a VIF threshold into the ALRSF (which has not been done in any of the logistic regression best subset selection MIP formulations presented in Section 2) serves two purposes:

1. Highly correlated variables are barred from being included in the final model in an effort

to reduce multicollinearity. It is well-known that correlated predictors add no additional information to the model (as potential signals are already contained within the other inputs) and result in unstable parameter estimates.

2. By eliminating some of the variables from the start, the logistic regression problem instance now has to consider fewer variables in its search for the best variable subset, which can potentially result in shorter execution times and fewer computational resources being consumed.

Next, allow $\beta'_j \in \mathbb{Z} \mid \forall j \in p$ to be an integer representation of the continuous decision variable $\beta_j \in \mathbb{R} \mid \forall j \in p$. A constant H is used to facilitate the conversion of β'_j from an integer variable to a continuous variable β_j by setting $\beta_j = \beta'_j/H$. A value for H can be chosen arbitrarily, noting that larger values for H will result in greater accuracy when estimating β, but increases the solution search space (resulting in extended execution times). The choice of H also influences the decimal precision of β. The proposed transformation allows the unique block structure of the MIP problem to be exploited such that it can be divided between a master and a sub-problem for Benders decomposition.[5] A brief overview of Benders decomposition will follow later on in this section. When the default Benders decomposition algorithm is considered, we note that all integer decision variables are allocated to the master problem, while continuous variables are assigned to the sub-problems. By applying the standard Benders decomposition algorithm to the MIP formulation proposed by Venter (2020), all decision variables apart from the cardinality selection variables z_j will be allocated to the sub-problems which will result in an invalid decomposition. By using integer values for the continuous regression parameters, we are able to make use of a valid Benders decomposition in an attempt to increase computational efficiency. The complete proposed MIP formulation for the piecewise linearised log-likelihood function (with the aforementioned modifications) is presented as

$$\max_{\beta',z,\Lambda} \sum_{i\in n}\sum_{k\in K}(\gamma_k Y_i - \log[1+\exp(\gamma_k)])\Lambda_{ik}, \tag{36}$$

subject to

$$\sum_{k\in K}\Lambda_{ik}\gamma_k = \frac{\beta'^T}{H}X_i \quad \forall i \in n, \tag{37}$$

$$\sum_{k\in K}\Lambda_{ik} = 1 \quad \forall i \in n, \tag{38}$$

$$0 \leq \Lambda_{ik} \leq 1 \quad \forall k \in K; \forall i \in n, \tag{39}$$

$$\frac{\beta'_j}{H} = 0 \quad \forall j \in p', \tag{40}$$

$$\left|\frac{\beta'_j}{H}\right| \leq Mz_j \quad \forall j \in p, \tag{41}$$

$$\sum_{j\in p} z_j \leq c. \tag{42}$$

[5]The use of Benders decomposition greatly reduces the memory requirement associated with solving the best subset selection MIP problem (formulated in Section 3) in comparison to using the standard branch-and-bound algorithm.

By incorporating constraint (40), we allow the model to automatically discard highly correlated variables that breach a VIF threshold supplied by the user (recall that $VIF_j > L, \ \forall j \in p'$). This also reduces the problem search space of the MIP. Note that constraint (37) serves as the connecting constraint that distinguishes between the master problem and sub-problem for Benders decomposition, where β'_j/H relates the integer regression parameters to the real-valued parameters β_j. The idea behind Benders decomposition is to fix a set of complicating variables to yield an optimisation problem that is notably easier to solve. A high-level introduction to the approach will be provided at the hand of a mixed integer linear programming (MILP) problem (note that the best subset selection logistic regression problem presented in (36)–(42) also classifies as a MILP). The objective function of the Benders decomposition problem can be defined according to

$$\min_{x,y} \rho^T y + \sigma^T x, \tag{43}$$

subject to

$$Wy = \kappa, \tag{44}$$

$$Oy + Dx = \tau, \tag{45}$$

$$x \geq 0, \quad \forall x \in \mathbb{R}, \tag{46}$$

$$y \geq 0, \quad \forall y \in \mathbb{Z}, \tag{47}$$

where $W \in \mathbb{R}^{m_1 \times n_1}$ represents a known coefficient matrix of size $m_1 \times n_1$ and $\kappa \in \mathbb{R}^{n_1}$ a pre-determined vector of size n_1 (Rahmaniani et al., 2017). The variables $x \in \mathbb{R}$ and $y \in \mathbb{Z}$ are continuous and integer decision variables, respectively. Let $O \in \mathbb{R}^{m_2 \times n_1}$ and $D \in \mathbb{R}^{m_2 \times n_2}$ denote matrices of size $m_2 \times n_1$ and $m_2 \times n_2$, respectively. Similar to κ, let $\tau \in \mathbb{R}^{n_2}$ denote a predefined vector of size n_2 (Rahmaniani et al., 2017). Finally, let σ and ρ denote the cost associated with each decision variable in x and y, respectively, with the objective function in (43) aimed at minimising the overall cost with respect to x and y. The integer decision variables y can be considered as being a set of complicating variables (Rahmaniani et al., 2017). Constraint (45) acts as a connecting constraint which relates the continuous decision variables x to the complicating integer decision variables y. The same can be said for constraint (37) in the ALRSF formulation, where (37) can be rewritten such that $\sum_{k \in K} \Lambda_{ik} \gamma_k - (\beta'^T/H) X_i = 0 \ \ \forall i \in n$ resembles constraint (45) in a standard MILP. The right-hand side of (37) can be rewritten such that it represents the standard form of the Benders decomposition problem. For any fixed value of y (defined as y^*) the original optimisation problem (43)–(47) can be defined as

$$\min\{\rho^T y^* + \min_{x \geq 0}\{\sigma^T x \ \mid \ Dx = \tau - Oy^*\}\}. \tag{48}$$

Since the inner minimisation problem is a continuous linear problem, it follows from duality theory that the inner problem can be dualised by adding dual variables θ_v to the relaxed constraint set given by $Dx = (\tau - Oy^*)$ where $v \in V$ is a set of extreme points. (Rahmaniani et al., 2017). Each solution to the inner maximisation problem in (49) generates one of the extreme points $\theta_v, \forall v \in V$. The dual representation of (48) then aims to

$$\max_{\theta_v, v \in V} \theta_v^T (\tau - Oy^*), \tag{49}$$

subject to

$$\theta_v^T D \leq \sigma \quad \forall v \in V. \tag{50}$$

It follows that (49) can be substituted into the second term of the minimisation problem presented in (48) which leads to a minmax optimisation problem equivalent to

$$\min\{\rho^T y^* + \max_{\theta_v, v \in V}\{\theta_v^T(\tau - Oy^*)\}\}, \tag{51}$$

subject to

$$\theta_v^T D \leq \sigma \mid \forall v \in V. \tag{52}$$

The feasibility of the inner maximisation problem is dependent on θ (as captured by constraint (52) above) but independent of the choice of y^*. According to Rahmaniani et al. (2017), the choice of y^* can, however, result in either a feasible or unbounded solution of the optimisation problem (Rahmaniani et al., 2017). Considering the latter case, there exists some direction r_g (with $g \in G$ describing all possible directions of unboundedness) for which $r^T(\tau - Oy^*) > 0$. To prevent unboundedness, a feasibility cut-off of the form

$$r^T(\tau - Oy^*) \leq 0, \tag{53}$$

is added to (51) which leads to

$$\min\{\rho^T y^* + \max_{v \in V}\{\theta_v^T(\tau - Oy^*)\}\}, \tag{54}$$

subject to

$$r_g^T(\tau - Oy^*) \leq 0 \mid \forall g \in G. \tag{55}$$

The objective function in (54) can be linearised by replacing the dual maximisation problem (second term of (54)) with a decision variable ξ. This linearised form of the optimisation problem (54)–(55) is known as the Benders decomposition master problem (MP) and is defined as

$$\min_{y,\xi} \rho^T y + \xi, \tag{56}$$

subject to

$$Wy = \kappa, \tag{57}$$

$$r_g^T(\tau - Oy) \leq 0 \mid \forall g \in G, \tag{58}$$

$$\theta_v^T(\tau - Oy) \leq \xi \mid \forall v \in V, \tag{59}$$

$$y \geq 0 \mid y \in \mathbb{Z}. \tag{60}$$

The Benders decomposition sub-problem (SP) is captured by (49)–(50). Ultimately, the MP aims to find suitable values for y and ξ, after which the SP is solved by using a value of y that is generated when the MP yields $y = y^*$. Constraints (58) and (59) can be referred to as feasibility and optimality cuts, respectively. The cuts are generated based on the solution of the sub-problem. If the solution of the sub-problem is unbounded, a feasibility cut of the form (58) is added to the MP. Similarly,

if the solution is both feasible and bounded, an optimality cut of the form (59) is added to the MP. This iterative solve-and-cut approach continues until the algorithm converges to a solution. The generated cuts prune the solution search space such that the overall computational burden of the algorithm is reduced. The MP will generally contain all of the integer variables, while the remainder of the variables (which are the continuous variables) will form part of the SP (Taşkin, 2010). The Benders decomposition algorithm can therefore be applied to the logistic regression MIP presented in (36)–(42) because it can be decomposed into an MP and SP as a result of the integer transformation that is proposed for the regression parameters.

4. Novel automated logistic regression solution framework (ALRSF)

The ALRSF's goal is to solve the MIP presented in (36)–(42) to optimality while also producing high-quality solutions. This is no easy task given that the best subset selection problem is NP-hard. As such, this section will clearly outline the adopted approach which includes a combination of mixed integer programming, Benders decomposition and heuristic-based techniques. Let $t \in T = \{1, 2, ..., |T|\}$ denote an outer looping constant that is used to keep track of the number of iterations that are executed by the ALRSF, where T is equal to p, the number of independent variables under consideration in the training set. Next, assume that all of the independent variables are sorted according to their information value (IV) in descending fashion (where the variable with the highest IV is ranked first and the variable with the lowest IV is ranked last) and IV_j denote the ranking of the jth variable relative to the other variables. The ranking dictates the order in which the independent variables are included in the solution algorithm at each step S. IV is widely used within credit scoring applications to assess an independent variable's predictive power and its ability to differentiate between the discrete target variable classes (Kolacek and Rezac, 2010; Kvesic and Gordana, 2012). In general, a variable with an IV that is higher than 0.2 or 0.3 is an acceptable candidate for inclusion in a scorecard model (Kolacek and Rezac, 2010; Kvesic and Gordana, 2012). For the ALRSF, however, the IVs are only used to rank-order each independent variable (relative to the other variables) with respect to its perceived predictive power. This means that the absolute values of the IVs themselves have no meaning.[6] Let $\Delta\gamma_k$ denote the distance between two splines $[\gamma_{k+1} - \gamma_k]$ associated with the piecewise linear approximation of the log-likelihood function (depicted in Figure 2). In case of symmetric splines $\Delta\gamma_k$ can be simplified to $\Delta\gamma$. Let $\Omega_i \in n$ represent a grid range evaluation variable which is used to identify whether the selected regression parameters cause the current grid range $[-\Gamma, \Gamma]$ to be inadequate and is calculated as $\Omega_i = (\beta'/H^T)X_i \;\; \forall i \in n$. To illustrate, refer to Figure 2 where the extreme points of the grid are shown as $-\Gamma = \gamma_0$ and $\Gamma = \gamma_k$. The solution search space is deemed to be too small when $\Omega_i \geq \Gamma$ or $\Omega_i \leq -\Gamma$, which implies that the grid with range $[-\Gamma, \Gamma]$ must be expanded to ensure that all possible solutions are considered by the logistic regression MIP. Specifically, the size of the grid needs to be expanded by some value ψ such that the grid points now range between $[-\Gamma - \psi, \Gamma + \psi]$. The constant ψ is arbitrarily selected and influences the incremental size with which the search space is increased. A large value for ψ may reduce the number of iterations needed to identify a sufficiently large grid, but can potentially cause the search

[6]For the purpose of this article IV was used as a measure of variable importance, however, any appropriate variable importance statistic (e.g. univariate Gini coefficient) can be utilised in place of IV to establish the independent variable importance rank-order.

space to be larger than needed, which can increase execution time. A balance should thus be struck between execution speed and the tightness of the solution search space. Let $\zeta \in \{0, 1\}$ represent a binary variable which is used to identify whether the grid range $[-\Gamma, \Gamma]$ is sufficiently large or needs to be expanded. When $\zeta = 1$ the current iteration t will be re-executed with an increased grid range. This will continue until the grid range is sufficiently large and represents the complete search space before the ALRSF can move to iteration $t + 1$. When $\zeta = 0$, the grid range considered is deemed to be large enough and the relevant performance measures and solution progress recorded for use in iteration $t + 1$.

A detailed study by Venter (2020) showed that a large value for K (the number of piecewise linear splines used to approximate the non-linear log-likelihood) can dramatically increase the execution time of the algorithm, while smaller K values result in an underestimation of the non-linear log-likelihood function. The aforementioned underestimation is, however, only marginal (and at times negligible) for small values of K. In general, models exhibited similar predictive performance across multiple empirical experiments for a wide range of K values. Let P_t denote the performance measure recorded at iteration t based on a holdout test set[7]. The performance measure P_t is used to determine whether the cardinality of the model (denoted by c) and set of regression parameters associated with the best model identified thus far (denoted by β'^b) should be updated with the solutions (the variables selected and their associated parameters) generated in the latest iteration. Let P_b denote the best value of P_t that is recorded over all iterations such that $P_b = \max(P_t)$ for $t = \{1, 2, 3, ..., |T|\}$. To evaluate the ALRSF, the area under the curve (AUC) was used as a performance measure P_t (although, theoretically, any model fit statistic such as the Gini coefficient or accuracy can also be used). A list of possible performance measures for binary classification problems is available in Table 1 of Sokolova and Lapalme (2009). Since the ALRSF starts with an iterative cardinality selection step and uses a performance measure P_t to assess the relative increase/decrease in model performance based on the inclusion of the jth independent variable, some form of performance tracking is required to identify when the predictive power of the model starts to plateau in spite of the addition of more independent variables. Specifically, when the performance of the model on the test dataset either reaches a plateau or decreases at iteration t of the ALRSF, it indicates that the cardinality parameter c (which dictates the number of variables to include in the model) can be set equal to the size of the subset identified at iteration $t-1$. The variable $u \in \mathbb{Z}^+$ is used as the aforementioned tracking variable that keeps track of the number of consecutive iterations for which the performance measure P_t has not improved on the best-recorded performance measure P_b. The value of u is then evaluated against a predefined threshold U, which determines the maximum allowable consecutive iterations for which P_b does not improve. When the threshold U is exceeded the cardinality step will be terminated and the search space pruning step initiated.

A holistic overview of Algorithms 1 to 5 will be provided next, which are explained in conjunction with Algorithm 1. This is because Algorithm 1 serves as an overarching driving algorithm that controls the various steps $S = \{0, 1, 2\}$ contained within the ALRSF. Since Algorithm 1 functions as the driving force behind each step of the ALRSF, the reader can continuously return to Algorithm 1 to identify the subsequent step that should be executed. In other words, the algorithmic flow will

[7] Note that the ALRSF evaluates the performance of the model based on a test set so that models that overfit on the training data are not presented as the champion.

jump between Algorithm 1 and the remainder of the Algorithms 2 to 5 as required by the ALRSF logic. Next, we discuss the flow of each algorithm contained within Algorithm 1:

- At the start of Algorithm 1, the step counter is set equal to $S = 0$ and we start at iteration $t = 1$. When the for-loop is entered at $t = 1$, the logic contained within Algorithm 2 is initiated.
- As we enter Algorithm 2 at iteration $t = 1$ and step $S = 0$, the pseudo code in lines 6 to 12 is executed. The aforementioned pseudo code identifies which regression parameters contained within the vector β' should be set equal to zero and which are allowed to take on integer values. Once this is done, we return to Algorithm 1 as per line 25 of Algorithm 2.
- Entering Algorithm 1 for a second time at iteration $t = 1$ and step $S = 0$ invokes the if statement in line 4 of Algorithm 1. This if statement instructs the solver to now consider Algorithm 3.
- The goal of Algorithm 3 is to automatically determine the cardinality parameter c (which dictates the size of the best subset of predictor variables) by iteratively adding independent variables one-by-one to the logistic regression MIP in (36)–(41) (note that constraint (42) is omitted in this step as we have not determined a final value for c yet). The change in the performance measure P_b is also assessed during this step. The final value of c is determined at the point where no further improvement in P_b has been observed for U consecutive iterations. At each iteration t, the MIP will abide by an additional constraint which dictates that $\sum_{j \in p} z_j = t$ based on the regression parameters that were allowed to be non-zero in Algorithm 2. Note that Algorithm 3 does not have an outer loop counter that corresponds to each iteration $t \in T$. Therefore, Algorithm 3 is called as needed (and non-iteratively) by Algorithm 1.
- We now direct our attention to the two for-loops in lines 7 to 9 and 13 to 20 of the pseudo-code of Algorithm 3. The first for-loop removes highly correlated variables from the MIP formulation by setting their regression parameters equal to zero. It is assumed that the VIF associated with each variable and the VIF threshold is provided by the user. The second for-loop determines if the sum of the regression parameters and independent variable values of each record falls within the range $[-\Gamma, \Gamma]$ that was specified for grid points used to linearise the log-likelihood (see lines 13 to 20). If this is not the case, the range of the grid is increased to ensure that the solution search space is sufficiently large and the tth iteration is re-executed.
- Once Algorithm 3 concludes in iteration $t = 1$, we revert back to Algorithm 1, as indicated by line 45 of Algorithm 3. Since the cardinality selection step has not reached the consecutive performance improvement limit U as described in lines 35 to 38, the step counter S is not incremented and as such will remain fixed at $S = 0$.

Algorithm 1 – ALRSF algorithmic driver.

$S = 0$
for $t = 1$ to T **do**
  Execute Algorithm 2 (regression parameter fixing heuristic) based on the value of S.
  if $S = 0$ **then**
    Execute Algorithm 3 (cardinality selection step) where $S = 0$.
  else if $S = 1$ **then**
    Execute Algorithm 4 (search space pruning step) where $S = 1$.
  else if $S = 2$ **then**
    Execute Algorithm 5 (optimality step) where $S = 2$.
  end if
end for
Solution Algorithm Completed.

Algorithm 2 – Regression parameter fixing heuristic.

if $(t \geq T)$ & $(S = 1)$ **then**
  $S = S + 1$
  $t = T - 1$
end if
if $S = 0$ **then**
  for $j = 1$ to p **do**
    if $(IV_j \leq t)$ **then**
      β'_j = free
    else
      $\beta'_j = 0$
    end if
  end for
else if $S = 1$ **then**
  for $j = 1$ to p **do**
    if $(|\beta'^{t-1}_j| > 0)$ or $(IV_j = t)$ **then**
      β'_j = free
    else
      $\beta'_j = 0$
    end if
  end for
else if $S = 2$ **then**
  for $j = 1$ to p **do**
    β'_j = free
  end for
  Return to Algorithm 1 - (driver algorithm).
end if

Algorithm 3 – Cardinality selection step – $S = 0$.

$c = 0$
P_t and $P_b = 0$
${\beta'}^b = 0$
$k = 0$
$u = 0$
$\zeta = 0$
for $j = 1$ to p' **do**
  $\beta'_j = 0$ if $VIF_j > L$
end for
Each regression parameter β'_j associated with variable j for which $VIF_j > L$ is set equal to zero.

Solve logistic regression MIP presented in (36)–(40) using Benders decomposition.
Cardinality Constraints (41) and (42) are excluded from the MIP.
for $i = 1$ to n **do**
  $\Omega_i = 0$
  $\Omega_i = \Omega_i + {\beta'}^T X_i$
  if $(\Omega_i \leq -\Gamma)$ or $(\Omega_i \geq \Gamma)$ **then**
    $|\Gamma| = |\Gamma| + \psi$
    $\Delta\gamma = \frac{2*|\Gamma|}{K}$
    $\zeta = 1$
  end if
end for
if $\zeta = 0$ **then**
  Calculate and record training and test performance measures P_t.
  Record ${\beta'}^t$.
  if $P_t > P_b$ **then**
    for $j = 0$ to p **do**
      ${\beta'_j}^b = \beta'_j$
    end for
    $P_b = P_t$
    $c = c + 1$
    $u = 0$
  else
    $u = u + 1$
  end if
  if $u = U$ **then**
    $S = S + 1$
    $t = 0$
  end if
else
  $t = t - 1$
end if
Cardinality c is produced as a result of this algorithm.
Save MIP solution found at iteration t for use as warm-start during the execution of iteration $t + 1$.
increment $t = t + 1$.
Return to Algorithm 1 - (driver algorithm).

- The ALRSF will once again initiate Algorithm 2 at iteration $t = 2$ and step $S = 0$ so that it can identify the next sequence of regression parameters that are allowed to take on non-zero values, after which a version of the logistic regression MIP is solved such that a model is fitted that contains t independent variables. When the stopping criterion in lines 35 to 38 of Algorithm 3 is met, the step counter S is incremented and the iteration counter is reset such that $S = 1$ and $t = 0$. We then return to Algorithm 1.

- A step value of $S = 1$ and iteration counter value of $t = 0$ in Algorithm 1 indicates that the ALSRF should execute the portion of the parameter fixing heuristic algorithm (Algorithm 2) that applies when $S = 1$. This is contained in lines 13 to 20 of the pseudo-code for Algorithm 2. The logic is quite similar to that of lines 5 to 12 with only a few minor adjustments. Once the appropriate regression parameters have been set equal to zero or non-zero, respectively, the ALRSF returns to Algorithm 1 (as indicated by line 25 of Algorithm 2).

- Next, the ALRSF moves to lines 6 and 7 of Algorithm 1. In line 7, the search space pruning step is initiated, the logic of which is found in the pseudo-code of Algorithm 4. Similar to Algorithm 3, Algorithm 4 contains no iteration counter and is therefore called as and when needed by Algorithm 1 (based on the step value S).

- The only difference between Algorithm 3 and Algorithm 4 relates to the fact that the logistic regression MIP is now solved by incorporating the cardinality constraint in (42) in Section 3. When $S = 1$, the number of independent variables that are allowed to enter the final model at each iteration is allowed to be at most $c + 1$. This cardinality constraint is enforced by the regression parameter fixing logic contained within Algorithm 2 where $S = 1$.

- Furthermore, Algorithm 4 also determines if the solution search space is sufficiently large enough by applying the logic contained within lines 10 to 18. If this is not the case, the grid that is used to linearise the log-likelihood is enlarged by adjusting the range as in line 14, after which iteration t is re-executed. In essence, Algorithm 4 attempts to iteratively include independent variables into the final subset of predictors (one by one in a sequential manner, where the sequence depends on the variables' IV) to determine if model performance can be improved when compared to a model that only contains the predictors that are already in the best subset of variables.

- Algorithm 4 will continue to execute until one pass through all of the independent variables has been completed, i.e. until iteration $t = T = p$ is reached. In doing so, the entire solution search space is pruned by solving sub-problems associated with the logistic regression MIP in (36)–(42). It is important to note that the solution produced by the model at iteration t is used as a warm-start for iteration $t + 1$ to ensure that any incremental progress made by continuous search space pruning is not lost. In other words, when the model prunes the search space during an iteration, the pruned space is recalled in future iterations in an attempt to reduce execution times.

- At iteration $t = T = p$, Algorithm 4 concludes and lines 1 to 4 of Algorithm 2 are invoked (as per Algorithm 1) which sets $S = 2$ and initiates the optimality step. Since the optimality step

Algorithm 4 – Search space pruning step – $S = 1$.

$\zeta = 0$
Cardinality parameter value c is generated by Algorithm 3.
for $j = 1$ to p' **do**
  $\beta'_j = 0$ if $VIF_j > L$
end for
Each regression parameter β'_j associated with variable j for which $VIF_j > L$ is set equal to zero.

Solve logistic regression MIP presented in (36)–(40) using Benders decomposition.
Cardinality Constraints (41) and (42) included in the MIP.
Cardinality is set equal to c.
for $i = 1$ to n **do**
  $\Omega_i = 0$
  $\Omega_i = \Omega_i + \beta'^T X_i$
  if $(\Omega_i \leq -\Gamma)$ or $(\Omega_i \geq \Gamma)$ **then**
    $|\Gamma| = |\Gamma| + \psi$
    $\Delta\gamma = \frac{2*|\Gamma|}{K}$
    $\zeta = 1$
  end if
end for
if $\zeta = 0$ **then**
  Calculate and record training and test performance measures P_t.
  Record β'^t.
else
  $t = t - 1$
end if
Save MIP solution found at iteration t for use as warm-start during execution of iteration $t + 1$
increment $t = t + 1$
Return to Algorithm 1 - (driver algorithm)

Algorithm 5 – Optimality step – $S = 2$.

Cardinality parameter value c is generated by Algorithm 3.
for $j = 1$ to p' **do**
  $\beta'_j = 0$ if $VIF_j > L$
end for
Each regression parameter β'_j associated with variable j for which $VIF_j > L$ is set equal to zero.

Use last MIP solution found in iteration t as warm-start to step $S = 2$.
Allow remaining regression parameters to be non-zero.
Solve logistic regression MIP presented in (36)–(40) using Benders decomposition.
Record final regression parameters, performance measures and optimality gap.
Return to Algorithm 1 - (driver algorithm).

is not iterative, t can be set equal to $t = T - 1$ considering that $S = 2$ is the last step in the ALRSF.

- Lines 21 to 24 of Algorithm 2 are subsequently executed when $S = 2$ and $t = T - 1$. This allows all the integer versions of the regression parameters in the vector β' to take on non-zero values (if deemed significant by the model) since the entire solution search space needs to be evaluated to provide a guarantee of optimality. Once all regression coefficients are allowed to take on non-zero values, the ALRSF returns to Algorithm 1.

- With $S = 2$, lines 8 and 9 in Algorithm 1 are executed, which initiates the optimality step. Algorithm 5 is one of the more simplistic components of the ALRSF and simply solves the logistic regression MIP in (36)–(42), where the cardinality parameter c is specified based on the results generated by Algorithm 3 and any regression coefficient is allowed to take on non-zero values. Ultimately, Algorithm 5 solves the original best subset selection regression problem, while the other Algorithms and steps presented thus far assist in automatically determining the cardinality parameter and pruning the solution search space in an attempt to reduce memory requirements and provide an optimality guarantee.

- The ALRSF concludes when optimality is guaranteed, an out-of-memory exception occurs or an alternative stopping criterion is met.

Now that the algorithmic flow of the ALRSF has been presented, some of the key elements associated with each of the three algorithmic steps $S = \{0, 1, 2\}$ will be discussed in more detail. The goal of the regression parameter fixing heuristic logic in Algorithm 2 (which is executed at steps $S = 0$ and $S = 1$ of the framework) is to reduce the computational memory burden associated with solving the complete logistic regression MIP in (36)–(42). From Algorithm 2, observe that the logic of the parameter fixing heuristic is adjusted based on the value of S. At step $S = 0$, Algorithm 2 ensures that the independent variables are iteratively included in the model based on their relative IV rank-ordering (starting with the most predictive variables first). This implies that the regression parameters associated with the most predictive variables are iteratively allowed to be non-zero, while the remaining coefficients are set equal to zero. This allows the user to ascertain the influence of each independent variable on model performance before the cardinality step at $S = 0$ is terminated (based on the threshold U) or before the next independent variable is considered for inclusion. As stated previously, during the execution of the cardinality selection step, the total number of independent variables that are allowed to have non-zero regression parameters at any given iteration will be equal to t.

Similar to before, the independent variables are once again added to the model in an iterative manner when $S = 1$ in Algorithm 2. However, the regression parameters of all the independent variables that were added to the best predictor subset in step $S = 0$ are now allowed to be non-zero and take on any real value. When $S = 1$, it is important to note that the number of variables considered for inclusion in the model will be equal to $c + 1$ during the execution of iteration t. However, at the end of iteration t, a maximum of c predictors may be present in the model. In essence, when an additional independent variable is considered for inclusion in the model in the current iteration t (given that there may or may not already be variables present in the model), one of the following can occur:

1. the variable is added to the final model if the number of non-zero regression parameters is less than c and the performance of the model is improved,

2. the variable replaces one of the other predictors that are already present in the model if the number of non-zero regression parameters is equal to c and if model performance is improved, or

3. if model performance does not improve (or deteriorates), the variable is rejected.

The solution search space is therefore iteratively reduced by testing the various regression parameter combinations. When $S = 2$, the final step of the parameter fixing heuristic (Algorithm 2) permits any regression parameter to be non-zero so that the entire solution search space is evaluated and an optimality gap is calculated. The solution from step $S = 1$ is used as a warm-start in step $S = 2$. This significantly reduces the size of the search space that needs to be considered before optimality can be proven.

Next, we consider Algorithm 3, which automatically determines the most appropriate cardinality parameter c based on the predictive power of the independent variables, the number of observations and performance of the model on the test set. In conjunction with the regression parameter fixing heuristic, Algorithm 3 also includes an additional functionality to allow for dynamic adjustment of the grid range $[-\Gamma, \Gamma]$ as explained earlier in this section. At each iteration t, the size of the grid range is evaluated to ensure that it is suitable and allows for the complete search space to be considered when a new independent variable is added to the model. The grid is updated by adding ψ to end-points such that the new range is $[-\Gamma - \psi, \Gamma + \psi]$ if the size of the grid is found to be insufficient (based on the value of Ω_i). This process of check-and-expand continues until the solution search space is sufficiently large. It should also be noted that the size of the grid will depend on the size of the regression parameters, the scale of each of the predictors and the cardinality parameter c. These factors may differ from one problem to the next and as such the proposed ALRSF allows for dynamic adjustment of the solution search space when required.

As part of Algorithm 3, the cardinality constraints presented in the logistic regression MIP in (36) -(42) are excluded from the model fitting step since the influence of each independent variable is evaluated without enforcing any restriction on the regression coefficients. The main output produced as part of the cardinality step is the final cardinality parameter c which will be utilised in steps $S = 1$ and $S = 2$, respectively. During each iteration, the MIP is solved by using the standard Benders decomposition approach (as discussed in Section 3). Furthermore, the integer solution found at iteration t is used as a warm-start in iteration $t + 1$. This prevents the model fitting algorithm from continuously resolving the same complete logistic regression MIP instance, which in turn should result in reduced computational memory demand and execution times.

When Algorithm 4 is invoked, the logistic regression MIP contained within each search space pruning iteration $t \in T = p$ is once more solved using Benders decomposition. The main difference between Algorithm 3 (executed at step $S = 0$) and Algorithm 4 (executed at step $S = 1$) is the inclusion of cardinality constraints in Algorithm 4, which limits the number of variables that can enter the model and facilitates best subset selection. The addition of the cardinality constraints to the MIP formulation introduces further complexity to the model, which leads to larger memory requirements because the model is tasked with filtering through numerous independent variable combinations.

This is where the search space pruning step is most useful as it iteratively prunes the search space by gradually testing the possibility of each independent variable entering the best variable subset. Algorithm 4, therefore, solves multiple sub-problems that only contain a subset of the available independent variables. As mentioned earlier, variables are iteratively added to the best subset (one at a time based on their IV rank-ordering) for testing and the Algorithm will only terminate when all independent variables have been tested at least once. Specifically, for Algorithm 4, the largest MIP that will be solved at any given point in time will be subject to the constraint $\sum_{j \in p} z_j \leq c + 1$, while the parameter c and iteration $t = 0$ is set equal to the value that was found in Algorithm 3. Similar to Algorithm 3, the search space pruning step also evaluates whether the size of the grid with range $[-\Gamma, \Gamma]$ is sufficiently large at iteration t. If found insufficient, the extreme points of the grid are increased by ψ until $\Omega_i \geq -\Gamma$ and $\Omega_i \leq \Gamma$ before continuing to iteration $t + 1$.

The final step of the ALRSF, which is executed by Algorithm 5 when $S = 2$, aims to improve on the solution generated by the search space pruning step by including all independent variables in the MIP solution process (except for those variables for which the VIF threshold was violated) while allowing their regression parameters β' to take on any integer value between the range $(-\infty, \infty)$. The best integer solution generated as part of step $S = 1$ using Algorithm 4 is used as a warm-start when executing Algorithm 5 in the final step where $S = 2$. This means that the optimality step $S = 2$ is not required to conduct a complete search of the entire solution space since a large portion of the search space has already been eliminated by Algorithm 4. The optimality step can therefore focus on improving the best integer solution while also attempting to prove optimality. The fact that the proposed solution framework is able to generate an optimality gap is a key advantage over normal heuristic-based variable selection algorithms, because it provides a guarantee on the optimality of the solution. As stated by Kutner et al. (2005), heuristic regression algorithms (such as stepwise selection, lasso and ridge regression) produce models that are "good enough". However, the user is not able to guarantee that the resulting model is the best possible model among all possible models and variable combinations. However, when an optimality gap is produced, the user is able to quantify a hypothetical distance between the predictiveness of the current model and the best possible model, with a gap of 0% indicating optimality. In other words, an optimality gap of 0% means that the final model (with at most c predictors) is the best possible model that could be obtained amongst all possible variable combinations.

4.1 Dummy example

We now attempt to clarify the three steps and five algorithms of the ALRSF using a practical logistic regression example that is based on a dataset containing six independent variables. It should be noted that the dummy example presented here is hypothetical and for illustrative purposes only. The performance and validity of the proposed ALRSF will be evaluated in Sections 5 and 6 using real-world datasets. The results of the dummy example are summarised in Table 1, which represents what a solution that is produced by the ALRSF would resemble. The first six iterations summarise the cardinality selection logic (Algorithm 3) while the following six relate to the search space pruning step (Algorithm 4). Finally, an attempt is made to prove optimality during the last iteration which relates to the optimality step described in Algorithm (5). Similar to what was presented in Algorithm 3, at iteration $t = 1$ the cardinality parameter is initialised to a value of $c = 0$. From iteration $t = 1$ to

Table 1. Novel ALRSF – Dummy Example.

Iterations	β'_1	β'_2	β'_3	β'_4	β'_5	β'_6	Test AUC	c	Gap %
Cardinality Step - (S = 0, c = 0)									
t = 1	Free	Fixed	Fixed	Fixed	Fixed	Fixed	51.34	1	–
t = 2	Free	Free	Fixed	Fixed	Fixed	Fixed	65.41	2	–
t = 3	Free	Free	Free	Fixed	Fixed	Fixed	87.60	3	–
t = 4	Free	Free	Free	Free	Fixed	Fixed	87.30	3	–
t = 5	Free	Free	Free	Free	Free	Fixed	87.31	3	–
t = 6	Free	Free	Free	Free	Free	Free	87.29	3	–
Search Space Pruning Step - (S = 1, c = 3)									
t = 1	Free	Free	Free	Fixed	Fixed	Fixed	87.60	3	–
t = 2	Free	Free	Free	Fixed	Fixed	Fixed	87.60	3	–
t = 3	Free	Free	Free	Fixed	Fixed	Fixed	87.60	3	–
t = 4	Free	Free	Fixed	Free	Fixed	Fixed	88.50	3	–
t = 5	Free	Free	Fixed	Fixed	Free	Fixed	88.65	3	–
t = 6	Free	Free	Fixed	Fixed	Free	Fixed	88.65	3	–
Optimality Step - (S = 2, $\forall \beta \in \mathbb{R}^p$, c = 3)									
t = 1	Free	Free	Free	Free	Free	Free	88.65	3	42.56

$t = 6$, the regression parameters[8] are sequentially added to the model based on IV rank-order.

For this example, we assume that the columns of Table 1 reflect the relative rank-ordering of the variables (based on their IVs) for $j = 1$ to 6. In other words, β'_1 corresponds to the variable with the highest IV, whereas β'_6 is associated with the variable with the lowest IV. Since the cardinality step tests the addition of one independent variable at a time, there will always be $c = t$ variables included in the model at iteration t. The regression parameters associated with the independent variables that enter the MIP for testing at each iteration are allowed to take on any integer value. This means that at iteration $t = 1$ regression parameter β'_1 is allowed to be non-zero such that $\beta'_1 \in \mathbb{Z}$, while the remaining regression parameters are fixed at zero. This essentially means that at iteration $t = 1$ a sub-problem with 6 independent variables is solved, utilising the standard Benders decomposition algorithm, with the regression parameter of one independent variable set allowed to be non-zero and the remaining parameters set equal to zero. Notice that the regression parameter β'_1 is allowed to be non-zero throughout the cardinality step (the first six iterations). This is because the variable associated with this parameter has the largest IV at iteration $t = 1$. Since variables are tested sequentially, the regression parameter β'_1 will always be allowed to be non-zero (throughout the first six iterations) as its predictor will have the highest IV, regardless of other variable combinations that are added. Furthermore, for iterations 1 to 6, note that the sequential inclusion of variables to the model means that one additional regression parameter is allowed to be non-zero at a time, as outlined by Algorithm 3.

[8] Each repression parameter $\beta'_j \ \forall j \in p$ is either set free or fixed. The term "Free" refers to the regression parameter $\beta'_j \in \mathbb{Z}$ being allowed to take on any integer value in $[-M, M]$, while "Fixed" relates to β_j being set equal to exactly zero.

Similarly, the test AUC P_t and highest recorded AUC P_b are both set equal to 0 when the cardinality step is initialised. At each iteration, P_t is recorded by solving the logistic regression MIP and fitting the model in a sequential manner (as discussed earlier). At iteration $t = 1$, P_1 is set equal to 51.34%, which is the AUC that was obtained at this step. Since P_1 is greater than the initial value of $P_b = 0$, we set $P_b = P_1$ and the cardinality parameter c is incremented by 1. The same is true for iteration $t = 2$ where the regression parameter β_2' is allowed to enter the model based on the IV rating of predictor variable X_2. A MIP with 6 independent variables is then solved, where the first two regression parameters are allowed to vary between $(-\infty, \infty)$, while the remaining regression parameters are set equal to zero. The test AUC P_2 after execution of the second iteration was found to be, 65.41% which is higher than P_b. As a result, we set $P_b = P_2$ and the cardinality parameter c is incremented once more such that $c = 2$. The same logic is applied for iterations 3 to 6 throughout the cardinality step (via the execution of Algorithm 3). At iteration 4, the test AUC that was found is smaller than P_b which means that the cardinality parameter c is not updated. In fact, for iterations 4, 5 and 6, none of the AUC values derived from the test set is greater than P_b, which means that the parameter c cannot be incremented. Ultimately, the cardinality step terminates at iteration 6 and the cardinality parameter c is set equal to the value of c that corresponds to the iteration where P_b was recorded. In this case, P_b was found at iteration 3 and we conclude that a cardinality parameter of $c = 3$ is most appropriate. This means that, for the specific logistic regression problem at hand, the best subset selection approach should produce the best 3-variable model.

By adding independent variables to the cardinality selection step in a sequential manner, the user is able to evaluate the incremental influence of each variable on performance of the model (as assessed on the test set so as to prevent overfitting). The first few variables that are added during the initial steps of the algorithm are expected to contribute the most to the predictive power of the model, as high IV values are assumed to be associated with high differentiation abilities. As independent variables are continuously added to the best variable subset, a point is reached where the model becomes saturated and the test AUC curve will either form a plateau (no improvement is obtained by adding additional variables to the model) or starts to drop off (adding additional variables to the model leads to overfitting). This implies that adding additional variables to the model beyond this point is therefore futile, which suggests that all remaining variables in the training set can be excluded. When the cardinality parameter remains constant for U iterations, the cardinality step can be terminated and the search space pruning step initiated. The cardinality parameter c can then be utilised in the pruning and optimality steps in conjunction with cardinality constraints, to enforce a limit on the number of independent variables that can be included in the final logistic regression model formulation. According to Kutner et al. (2005), the order in which variables are added to or removed from a model can exert a significant influence on the final model that is obtained. For example, during stepwise selection procedures, the variable that is added during iteration t (and its regression coefficient) is dependent on the variables that were included in the model at iteration $t - 1$. One might therefore think that the proposed ALRSF presented in this paper suffers from the same drawback due to the sequential addition of variables to the model during step $S = 0$. Note, however, that this is not the case. The iterative addition of variables to the model only takes place during the cardinality step and is simply used to determine the most appropriate value of the cardinality parameter c. During step $S = 2$ of the ALRSF, all combinations of variables are tested, which means that the order in which variables enter the model is no longer influential.

The search space pruning step $S = 1$ (which is facilitated by Algorithm 4) can be described by the second set of six iterations in Table 1. The main purpose of the pruning step is to reduce the solution search space, which ultimately reduced the optimality gap percentage in the final optimality step $S = 2$. This is done by iteratively testing independent variable combinations to determine which variable subset will improve the objective function value in (36) while conforming to constraint (42) in the logistic regression MIP. The total number of independent variables that are allowed to have non-zero regression parameters at any given iteration during the execution of step $S = 1$ may not exceed $c + 1$. For the particular example presented in Table 1, a total of four regression parameters are therefore allowed to be non-zero during the search space pruning step $S = 1$, as the most appropriate cardinality parameter was determined to be $c = 3$ during the cardinality step $S = 0$. This is evident when we consider the third and fourth iterations of the search space pruning step, where the ALRSF determined that allowing β'_4 to be non-zero instead of β'_3 yields an improvement in model performance. As a result, the predictor variable associated with β'_3 was removed from the best variable subset and replaced with the variable associated with β'_4. In other words, the independent variables are iteratively added to the model and are either allowed to enter the best subset by replacing one of the current variables with a non-zero regression parameter or are rejected, at which point the next iteration $t + 1$ is initiated with no change to the best subset. During the execution of the search space pruning step, note that the first three regression parameters are allowed to be non-zero during the first three iterations because the cardinality parameter was specified as $c = 3$ and the predictor variables associated with these coefficients had the highest IVs. The idea behind the search space pruning step is to challenge this assumption (that the first three variables produce the best-performing model) by iteratively adding another variable to the model and testing to determine if the added variable can replace one of the variables that are already included in the best variable subset. When $S = 1$, the first iteration tests whether the variable associated with β'_1 can improve the objective function when added to a subset that is allowed to contain at most three variables. However, since this variable is already part of the best subset and the size of the subset is smaller than $c = 3$, no change is recorded. The same can be said for the variables associated with β'_2 and β'_3 during iterations $t = 2$ and $t = 3$ of the search space pruning step. From Table 1, a change in the best subset is only visible at the fourth iteration of the search space pruning step since the variable associated with β'_4 improved the predictive performance of the model by replacing the variable associated with β'_3 in the best variable set. The same logic applies for every iteration contained in the search space pruning step, where we note that no significant improvement in model performance is recorded during iterations 5 and 6. Once all of the predictors have been tested at least once, the optimality step $S = 2$ can be initiated since more than one pass over the set of independent variables will not result in further improvements in the model's predictive performance.

The final step of the ALRSF is the optimality step (which is facilitated by Algorithm 5). The solution obtained during the final iteration of step $S = 1$ (search space pruning step) is used as a warm-start in the optimality step $S = 2$, which prevents the complete logistic regression MIP from being solved and leverages the reduced search space provided by the pruning heuristic. If the warm-start method was not utilised, the progress made in step $S = 1$ would be disregarded and step $S = 1$ would add no value to the solution framework. During step $S = 2$, all regression parameters are allowed to be non-zero to ensure that all possible independent variable combinations are considered by the logistic regression model and to determine the extent to which the final log-likelihood solution

deviates from optimality. The solution framework will only terminate when optimality has been proven, an out-of-memory error occurs or a predefined user-specific stopping criterion is met. The optimality gap presented in Table 1 is only hypothetical and shows that step $S = 2$ is capable of producing a measure of optimality for the final solution.

In summary, the proposed novel ALRSF provides a holistic approach towards fitting large-scale logistic regression models that are formulated as MIP problems to perform best subset selection. This is achieved by iteratively pruning the solution space and employing the default Benders decomposition approach (which is only possible if the model is formulated as the MIP in (36)–(42)) to find high-quality solutions with smaller memory requirements and providing an optimality gap for the final solution. The end-result is a proposed approach that aims to find high-quality solutions that impose less computational memory requirements, are executed faster than comparative formulations and provide a guarantee on the quality of the model in the form of an optimality gap.

5. Model verification

In this section, the proposed ALRSF is benchmarked against the standard branch-and-bound MILP formulation of the logistic regression best subset selection problem that was proposed by Venter (2020)[9]. Two small-scale datasets that contain six and eight variables, respectively, will be considered. The problems in question relate to default prediction for credit card customers and are solved using default data provided by Taiwanese credit providers that were published on the UCI machine learning repository. The original dataset consists of 24 independent variables and 30000 observations. For illustrative purposes, a training set of 200 observations is used. This is to ensure that both problems are solved such that optimality is guaranteed (i.e. an optimality gap of 0% is obtained) within an acceptable time frame so that a proper comparative study can be conducted. The binary target class was distributed such that 23% of the observations were assigned a value of one (default). Table 2 contains the descriptions of each of the independent variables that were considered for inclusion in the model. Three main aspects of the proposed solution framework are considered when the final models are evaluated:

1. The regression parameter estimates produced by each of the algorithms need to be inspected and compared in order to determine whether they are appropriate.

2. The best variable subset produced by each model needs to be examined to assess the variables that were selected.

3. The range of the grid that is used to linearise the non-linear log-likelihood (with extreme points $[-\Gamma, \Gamma]$) needs to be scrutinised to determine whether the solution space is sufficiently described.

For the verification of the grid range, the manual grid selection method proposed by Venter (2020) was utilised to obtain a suitable grid for the standard branch-and-bound algorithm. The ALRSF was

[9] Note that commercially available software (such as SAS) also makes use of variations of the standard branch-and-bound method to execute best subset selection within regression procedures. For more information, see Appendix C of Venter (2020) that summarises the Leaps and Bounds (LBA) method of Furnival and Wilson (2000) and the branch-and-bound algorithm (BBA) of Gatu and Kontoghiorghes (2006) which are used in SAS procedures.

Table 2. Independent variable description.

Independent variable	Variable description
1	Amount of credit extended to the client
2	Client gender
3	Level of education
4	Marital status
5	Client age
6	Repayment status in September 2005
7	Repayment status in August 2005
8	Repayment status in July 2005

developed in C++ and the MIP formulations were solved utilising the Cplex concert technology API. The experiments were conducted on a Ryzen Linux Mint machine with an AMD Ryzen 5 1600 6-core processor and 32 GB random access memory (RAM). Python (with Anaconda distribution) and Microsoft Excel were used for supplementary data processing, such as calculating the VIF and IV of each independent variable. The training and test sets were saved as Microsoft Excel comma-separated files (CSV) and used as input for both the standard branch-and-bound and ALRSF algorithms.

In Table 3, the regression parameters that were produced by the standard branch-and-bound algorithm and ALRSF are shown. It is evident that near-identical regression parameters are produced by both algorithms for two versions of the problem (one where six inputs are allowed to enter the model and one where eight inputs are allowed to enter the model). There are, however, some minor differences concerning the decimal precision that will be addressed later in this section. With reference to the objective function (which is the log-likelihood), an identical value was recorded for both algorithms. This provides confidence that the proposed ALRSF framework outputs accurate and trustworthy results that are in line with the more-established branch-and-bound approach (which serves as the default method for performing best subset selection).

Next, a set of experimental results are presented to explain the decimal differences observed in Table 3 by varying the integer conversion constant H in (40) which was formally introduced in Section 3. The integer conversion constant H, which facilitates the conversion between the integer regression parameters β' and continuous parameters β, is firstly selected to be a 1000 at first and then changed to 10000. This allowed the influence of H on the final parameter estimates to be evaluated. By comparing the results in Table 4 with those in Table 3, it is clear that the ALRSF parameter estimates are more closely aligned with those produced by the branch-and-bound algorithm when $H = 10000$. Therefore, a larger value for H ensures greater decimal precision and provides a better

Table 3. Regression parameter estimate comparison.

Logic	c	p	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	LogL
ALRSF	4	6	0.655	2.460	0	0	1.536	-0.350	-5.668	-	-	-98.63
B&B	4	6	0.654536	2.46055	0	0	1.53657	-0.34927	-5.66825	-	-	-98.63
ALRSF	3	8	0.466	2.347	0	0	1.691	0	-5.738	0	0	-98.67
B&B	3	8	0.465977	2.34726	0	0	1.69113	0	-5.73871	0	0	-98.67

Table 4. Variable conversion constant H – ALRSF.

H	c	p	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	OBJF
1000	4	6	0.655	2.460	0	0	1.536	-0.350	-5.668	-	-	-98.63
10000	4	6	0.65475	2.46032	0	0	1.53659	-0.34926	-5.66828	-	-	-98.63
1000	3	8	0.466	2.347	0	0	1.691	0	-5.738	0	0	-98.67
10000	3	8	0.46597	2.34726	0	0	1.69113	0	-5.73871	0	0	-98.67

Table 5. AUC performance measure comparison.

Logic	H	c	p	Train AUC	Test AUC
ALRSF	1000	4	6	71.56	74.36
ALRSF	1000	3	8	71.34	74.52
ALRSF	10000	4	6	71.56	74.35
ALRSF	10000	3	8	71.34	74.48
B&B	1000	4	6	71.56	74.35
B&B	1000	3	8	71.34	74.48

approximation of the true underlying regression parameters, but increases the size of the search space. The choice of H may also be problem-specific. It is important to consider the influence of the conversion constant H on the performance measures that are calculated from the training and test sets, respectively. A comparison of the AUC values that were achieved on the training and test sets, respectively, can be found in Table 5. Identical training AUC values were found for all experiments, regardless of the value of H, which might suggest that H has an insignificant impact on model fitting and training set performance. It is, however, interesting to note that when the test AUC is examined, the ALRSF with $H = 1000$ produces higher AUC values when compared to models produced by both the branch-and-bound algorithm and an ALRSF with $H = 10000$. This might be attributed to a marginal reduction in overfitting on the training set (because the regression parameters have lower decimal precision) which ultimately allows for slightly better generalisation performance. Nevertheless, the performance increase can be regarded as negligible. Ultimately, we can conclude that the proposed ALRSF produces results that are expected (when compared with default best subset selection techniques) and that the performance and generalisation capabilities of the model are not severely affected by the parameters of the proposed MIP.

Table 6 shows that the best subset consists of three independent variables when a logistic regression model is fitted to the eight-variable dataset using the ALRSF. The purpose of the results recorded in Table 6 is to evaluate the logic of the solution algorithm and to determine if a cardinality parameter of $c = 3$ is a sensible choice. Recall from Section 3 that a specific cardinality selection criteria can be set by the user to determine whether the improvement in the performance measure P_t between

Table 6. Cardinality selection verification.

Logic	c	p	U	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 8$
ALRSF	3	8	2	72.04	72.24	73.96	73.86	73.70	75.32	75.34	75.32

iterations is sufficient enough to allow the cardinality parameter c to be incremented with one. For the purpose of the model shown in Table 6, the selection criterion was specified such that the cardinality parameter can only increase when the performance measure of the current iteration, P_t, is larger than, the best performance measure P_b achieved this far. When the results of Table 6 are assessed, it is clear that the AUC increased consistently from iterations $t = 1$ to $t = 3$, which implies that the cardinality parameter was set equal to $c = 3$ at iteration $t = 3$. The consecutive increment limit U for this experiment was specified as $U = 2$, which means that the model will also consider the four-variable and five-variable models produced in iterations 4 and 5 before terminating. However, given that no improvement in the test AUC is recorded for iterations 4 and 5, the cardinality selection step terminates at iteration 3 and it is concluded that the most suitable size for the best subset is three independent variables. If the incrementation limit U was specified to be larger than two, the model would have identified improvements in the test AUC at iterations 6 and 7, which may have indicated that the most suitable value for the cardinality parameter should either be $c = 6$ or $c = 7$. In general, the user should consider a sufficiently large value for U to ensure that the best subset selected by the final model is not too small, sufficiently describes the problem space and that most of the variables that add to the predictive power of the model are considered. However, there also exists a trade-off when using a value for U that is too large, as inflated values for U will result in extended model run times. Model developers should also consider the possibility of an AUC plateau forming after a certain number of iterations, where the addition of more variables to the model does not lead to significant improvements in AUC. In such circumstances, a large U value will produce a cardinality set that is similar to one that is obtained when a small U is used.

6. Empirical results and discussion

In this section, the added benefits of the ALRSF over the standard branch-and-bound best subset selection problem are demonstrated by applying the proposed approach to realistically-sized real-world classification problems. All problem instances considered were obtained from the UCI machine learning repository and cover a wide range of industries including health care, engineering and entertainment.

6.1 Experimental Datasets

Three problem instances are considered, the first of which is the superconductivity regression problem. The dataset is made up of 21263 observations and 81 independent variables. The independent variables describe the attributes of various superconductors including density, atomic mass, electron affinity, fusion heat and thermal conductivity. The real-valued response variable Y_i was transformed into a binary target variable by specifying $Y_i^{bin} = I(Y_i \geq 1400)$ where $I(\cdot)$ is an indicator function.

Three experiments were performed for the transformed superconductivity classification problem. The 21263 observations were subdivided into three training sets with 6500, 8000 and 9000 observations, respectively, while predictive performance was assessed using a hold-out set of 6000 observations. The goal of these three experiments is to demonstrate the flexibility of the ALRSF when solving large problem instances that consist of a large number of independent variables and sizeable training sets. When literature on best subset selection using MIP formulations is reviewed, one will often find that models are either fitted on datasets that contain a large number of independent

variables or a large number of observations, but rarely both. One of the largest problem instances in literature was solved by Venter (2020), with 6000 observations and 58 independent variables that originated from the superconductivity dataset. There is therefore potential to consider a best subset selection problem (within the context of a MIP formulation) that contains a training set that is large with respect to both n and p.

The second problem instance considered is known as the relative location of CT slices regression problem, which describes the CT images for 74 patients. The dataset in question consists of 53500 observations and 385 independent variables. The target variable (location of CT scans) is an integer-type variable and ranges from 0 to 180, where 0 references the soles of the patient's feet and 180 the top of the head. For the purpose of solving classification problems, the integer target variable was transformed to a binary target by specifying $Y_i^{bin} = I(Y_i \geq 50)$.

From the 385 independent variables, a total of 18 unary variables were removed. A further 125 independent variables were removed from the training and test sets based on a VIF threshold of 10. As discussed in Section 4, a functionality was added to allow for the removal of highly correlated independent variables in an attempt to improve numerical stability and statistical interpretability of the final model produced by the framework (Akinwande et al., 2015). This resulted in a final set of 242 independent variables from which the ALRSF can identify the best subset of variables.

Similar to the superconductivity problem instance, three experiments were derived from the total set of available observations, with each training set consisting of 5000, 6000 and 6500 observations, respectively. A test set of 25000 observations was utilised for model evaluation. This problem instance is focused on testing the proposed ALRSF's ability to solve classification problems with a large number of independent variables while still allowing for a moderate number of training observations to be considered.

The third and final round of experiments was conducted by utilising the online news popularity dataset, which consists of 39797 observations. Three training sets comprising 11000, 12000 and 13000 observations, respectively, were then derived. Furthermore, a test set consisting of 24000 observations was used to assess model performance. From the 58 independent variables available, 13 were excluded from the training set based on the VIF exclusion criteria. The continuous target variable (number of shares) was modified to be suitable for a classification problem such that all observations were divided between two classes. The target variable Y_i was transformed by specifying $Y_i^{bin} = I(Y_i \geq 1400)$.The online news popularity classification problem is a moderately sized problem for which a large set of training observations were employed to further validate the proposed solution framework. Additionally, the appropriateness of the ALRSF within application domains that relate to different fields of interest is also explored by considering three separate datasets that relate to vastly different prediction problems.

6.2 Modelling assumptions

The main assumptions associated with the modelling framework (with reference to the experiments in this section) can be summarised as follows:

1. An execution time limit of 345600 seconds for all experiments, except for the CT slice location classification problem, was enforced. For the CT slice location problem, a total of 691200 seconds were allotted due to the size of the problem instance. The execution time of the

ALRSF is directly affected by the number of independent variables and observations.

2. To allow for a comparative study between the proposed ALRSF and the standard branch-and-bound algorithm, the problem-specific parameters (as identified by the ALRSF) are also used for the branch-and-bound algorithm. This includes the cardinality parameter c and the grid range $[-\Gamma, \Gamma]$. It is important to note that for the standard branch-and-bound algorithm, the latter would have to be determined by means of a manual selection procedure while the proposed ALRSF specifies an appropriate grid in an automated fashion.

3. A total of $K = 400$ splines (or line segments) were used to approximate the non-linear log-likelihood objective function. A detailed study with regards to the influence of the value for K can be found in Venter (2020).

4. Cplex only outputs a solution when the problem either terminates based on a predefined termination criteria or when the algorithm has proven optimality. When a problem instance fails (e.g. in the case of an out-of-memory error), a solution cannot be extracted from the Cplex C++ concert technology API and the problem will have to be re-executed with a predefined stopping criteria. This was the case for most branch-and-bound experiments against which the ALRSF is benchmarked. When an experiment failed on an out-of-memory exception, the instance is re-executed, but with a set time limit to allow the user to extract the best solution that was found before termination.

5. The integer conversion constant H (as discussed in section 4) is arbitrarily set equal to 100000 for the experiments considered in this section. In Section 5 we noted that the value of H has a negligible effect on the performance of the final model, which means that a value for H can be arbitrarily set by the user. A larger value for H will, however, better approximate the true underlying logistic regression parameters.

6. The grid point adjustment constant ψ was selected to be equal to 10. This is to facilitate a gradual increase in the grid range $[-\Gamma, \Gamma]$ while simultaneously reducing the number of iterations required to accurately discover the appropriate solution search space.

7. The incremental threshold constant U, was set equal to a value of 10 based on preliminary verification experiments. A value of $U = 10$ implies that the cardinality selection step can terminate when no improvement in model performance is recorded with the consecutive inclusion of up to 10 independent variables to the best variable subset (in addition to the predictors that are already contained within the best subset). The empirical results that follow seem to suggest that a value of $U = 10$ is reasonable, even for large datasets.

6.3 Empirical results

Table 7 and Figures 3 to 5 provide a summary of the results that were obtained for the experiments that were described in Section 6.1. Table 7 contains key assessment criteria, such as memory usage, optimality gap, train and test AUC and best integer solution, that can be used to compare the proposed ALRSF with the standard best subset selection branch-and-bound algorithm.

In general, the memory usage required by the proposed ALRSF is much lower than the standard branch-and-bound algorithm. In some cases, a significant reduction in memory of up to 10 GB is

observed. This, in turn, allows for larger datasets to be considered when attempting to fit a logistic regression model following a best subset selection approach. For the superconductivity classification problem with a training size of 9000 observations, the standard branch-and-bound algorithm was not able to find a solution as a result of insufficient memory (the reader will note that an entry of "ME" in Table 7 indicates that a "memory exceeded" exception occurred). In contrast, the ALRSF was able to find a high-quality solution with a memory buffer of 1.8GB from the available memory of 32 GB still available. Similarly, for both the CT slices and online news popularity classification problems (6500 and 13000 training observations, respectively), much shallower solutions that exist within the solution tree were found when utilising the standard branch-and-bound algorithm, while high-quality solutions were obtained when applying the ALRSF. This is further corroborated by critically evaluating the optimality gap and best integer solution obtained for these two datasets in Table 7. A significant improvement with regard to the optimality gap was observed, especially for the superconductivity problem.

When the train and test AUC is considered, it can be seen that the performance between the two algorithms is relatively similar for smaller datasets. The superiority of the ALRSF with respect to model performance is, however, evident when using the solution framework to solve large-scale problem instances. The algorithmic improvements made to the standard branch-and-bound algorithm (as described in Sections 3 and 4) clearly add to the flexibility and robustness of the solution framework. For the superconductivity problem, the test AUC obtained by the ALRSF ranged between 86.47% and 87.21%. A slight increase in test AUC is observed with an increase in the set of training observations, which may be attributed to more training data that are more representative of the underlying population. A relatively similar trend is observed for the CT location and news popularity classification problems. For the CT slice location problem, the test AUC increased from 89.04% to 94.34% with an increase in the number of training observations from 5000 to 6500 records. Finally, we note that the ALRSF was able to yield AUC values between 68% and 69% for the online news popularity dataset, which are comparable to AUC values of the models that were produced by the branch-and-bound algorithm.

It is also worth noting that the ALRSF is able to dynamically adjust the best variable set based on the available training data and the problem instance considered. The cardinality parameter c varied between experiments, which suggests that the solution framework is able to detect changes in trends contained in the underlying data and adjust the proposed model accordingly to best describe the problem instance. The cardinality selection criteria will influence the size of the final model selected. For the online news popularity problem, almost all of the independent variables were included in the final model. This is because of consistent slight increases in the test AUC that are recorded with the addition of each new variable, as seen in Figure 5.

A possible solution that will yield more parsimonious models might be to use a less stringent cardinality selection criteria which stipulates that the increase in the AUC must be larger than some predefined percentage before a variable is added to the model, e.g. the AUC must increase by 10% from iteration $t-1$ to iteration t before an additional variable is considered for inclusion.

The ability of the ALRSF to solve a problem instance with 242 independent variables and 6500 training observations is considered to be a noteworthy achievement for the best subset MIP problem formulation, given that standard branch-and-bound methods will likely fail when presented with such a large number of predictors and observations. Similarly, for the superconductivity classification

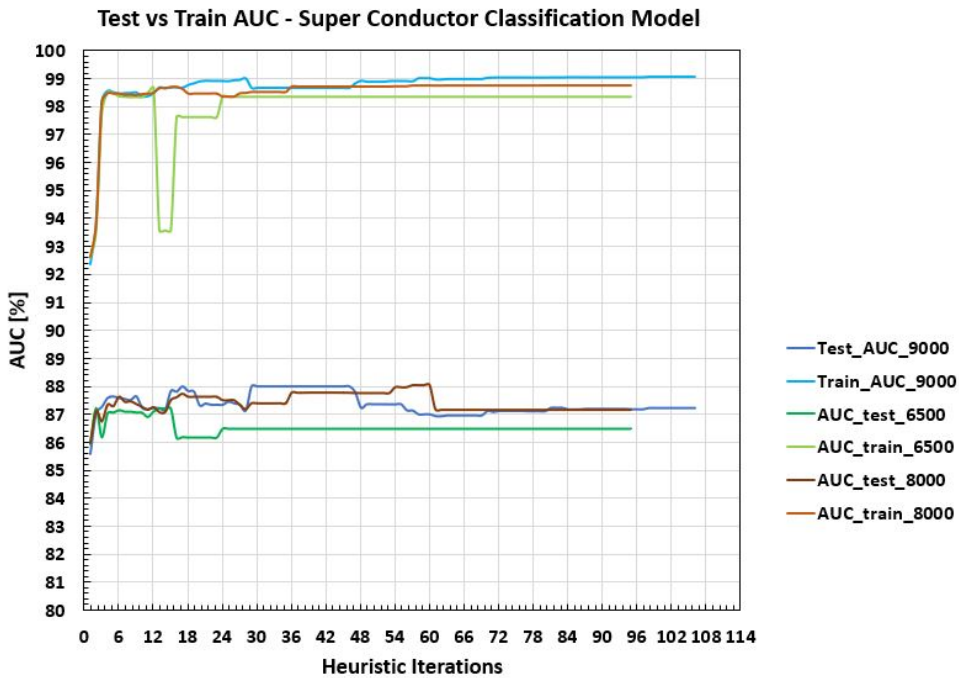

Figure 3. Superconductivity classification problem.

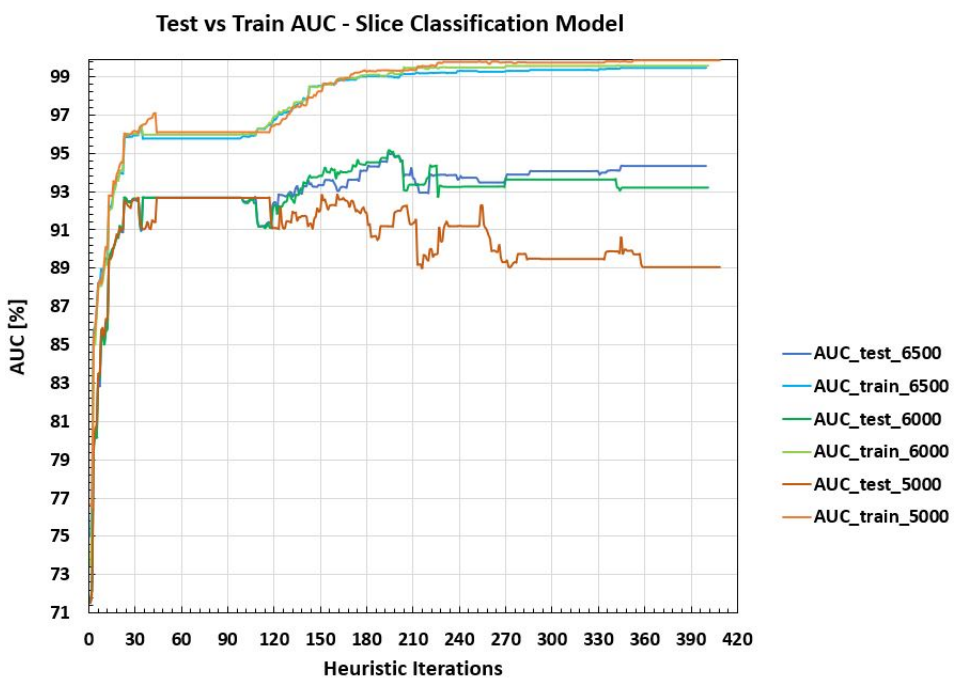

Figure 4. Relative CT slices location classification problem.

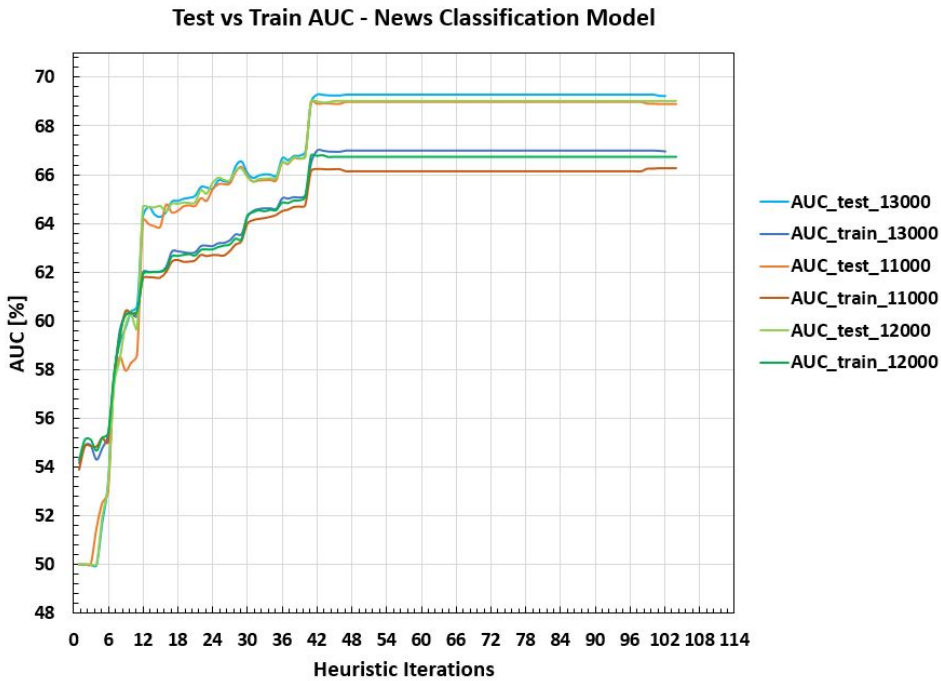

Figure 5. Online news popularity classification problem.

problem, a problem instance with 81 independent variables and 9000 training observations was successfully solved. The aforementioned solutions can be considered as some of the largest problems that were successfully executed within a best subset selection MIP framework (when compared to the available literature on similar formulations). The ALRSF pushes the boundaries concerning computational efficiency and increases the type and size of problems to which a best subset logistic regression solution framework can be applied. Figures 3 to 5 are a visual representation of the iterative solution process contained within the ALRSF that was described in Section 4. It is important to note that the iterative approach allows the user to continuously monitor the performance of the model as fitting progress, which in turn can also be used to detect overfitting on the training set. The cardinality parameter c for each experiment is listed in Table 7. Figures 3 to 5 illustrate the transition from the cardinality step $S = 0$ to the search space pruning step $S = 1$ when a value of $U = 10$ is added to the problem formulation. It is evident from the visual representations of the solution framework that adding more independent variables to the best variable subset will not necessarily lead to increased predictive performance. The aim of the solution framework is therefore to produce parsimonious models in an automated fashion that are still sufficiently predictive when applied to unseen data. Figure 4 clearly shows an instance of overfitting, where an increase in the training AUC is accompanied by a drop in the test AUC when 5000 training observations are considered. In general, the visual representation of the solution framework supports the notion that an increase in the amount of training data results in an increase in the model performance. Nevertheless, the ALRSF equips the user with analytical capabilities to terminate the solution framework at any given point in time when an acceptable solution is found, which can be used to address the problem of overfitting.

This means that the ALRSF is also more interpretable, controllable and transparent when compared to the standard branch-and-bound approach.

7. Conclusion and recommendations

A novel automated logistic regression solution framework was presented that allows for a holistic deterministic approach towards solving the best subset selection MIP problem formulation (presented in Section 3) of a binary logistic regression classification problem. The ALRSF is executed via a three-step approach, which involves the following:

1. The automatic specification of a cardinality parameter c and selection of the best variable subset based on model performance.
2. Pruning of the search space described by the MIP formulation, which is achieved via a Benders decomposition regression parameter fixing pruning algorithm.
3. The calculation of an optimality gap in order to provide a guarantee of optimality, which is not possible for heuristic variable selection methods (such as stepwise, lasso and ridge regression) that do not employ MIP formulations. Furthermore, the optimality step also attempts to improve the solution that was discovered by the pruning algorithm in the previous step.

Empirical results presented in Section 6 show that significant improvements in the final logistic regression solution are achieved when the ALRSF is applied instead of the standard branch-and-bound model. Specifically, the ALRSF allows for better solutions to be achieved for larger problems, requires less memory during execution, and, overall, obtains higher quality solutions with lower optimality gaps due to its efficiency and ability to effectively manage the solution search space. The ability to efficiently identify an optimal subset of independent variables from large-scale datasets that contain many observations and variables by using a deterministic solution framework may incentivise more users to consider exact methods for model-fitting exercises, as such methods are able to provide the user with a measure of optimality (which is not possible when using heuristic fitting algorithms). Another consideration is minimising the need for highly skilled resources to facilitate model development by employing a more automated framework.

The proposed solution framework naturally lends itself to other modelling techniques such as linear regression and even possibly neural networks. Future endeavours can, therefore, be concerned with the expansion of the proposed ALRSF to accommodate more than one model specification within an exact best subset selection MIP problem formulation. Expanding the proposed MIP formulation to allow for best subset selection within the context of multi-class classification problems can also be explored.

The search space pruning heuristic mainly focuses on reducing the lower bound of the MIP when formulated as a maximisation problem (which is the case for the log-likelihood function). As a possible area for improvement, the addition of Lagrangian relaxation during the optimality step ($S = 2$) to tighten the MIP upper bound can be considered. The cardinality constraints are seen as complicating constraints, the effect of which can be managed more appropriately by possibly including these constraints as a Lagrangian relaxation in the objective function. This might further reduce the optimality gap and the time required to prove optimality.

Table 7. ALRSF vs. B&B – experimental results.

Instance Identifier/ observations	Cardinality / Total independent variables (After VIF Exclusion) / VIF Excluded	Memory Used (Gb)	Integer Solution	Gap (%)	Training Accuracy (%)	Training AUC	Test Accuracy (%)	Test AUC
Superconductivity Classification Problem								
1-ALRSF/ 6500	2/81/0	25.4	-808.75	42.48	95.31	98.34	83.77	86.47
1-B&B/ 6500	2/81/0	>32	-1819.12	75.40	86.11	92.77	77.93	86.10
2-ALRSF/ 8000	6/81/0	28.1	-823.64	30.73	96.35	98.78	84.95	87.17
2-B&B/ 8000	6/81/0	>32	-2260.73	74.47	89.49	94.05	82.32	87.54
3-ALRSF/ 9000	17/81/0	30.2	-773.22	23.52	97.11	99.07	83.45	87.21
3-B&B/ 9000	17/81/0	>32	ME	ME	ME	ME	ME	ME
Relative CT Slice Location Classification Problem								
1-ALRSF/ 5000	32/242/125	26.4	-127.75	100.00	99.20	99.84	87.41	89.04
1-B&B/ 5000	32/242/125	>32	-1123.80	99.96	90.82	96.74	84.04	91.23
2-ALRSF/ 6000	23/242/125	31.1	-359.40	97.82	97.83	99.56	89.56	93.19
2-B&B/ 6000	23/242/125	>32	-4158.88	99.81	50.92	50.00	60.62	50.00
3-ALRSF/ 6500	23/242/125	>32	-439.70	95.92	97.45	99.47	90.16	94.34
3-B&B/ 6500	23/242/125	>32	-3558.65	99.69	77.32	81.79	71.82	77.22
Online News Popularity Classification Problem								
1-ALRSF/ 11000	41/45/13	20	-7100.34	OPT	63.06	66.25	63.63	68.89
1-B&B/ 11000	41/45/13	>32	-7107.75	0.11	62.90	66.13	66.13	68.89
2-ALRSF/ 12000	45/45/13	22	-7729.86	OPT	62.94	66.75	64.91	69.01
2-B&B/ 12000	45/45/13	30	-7729.86	OPT	62.94	66.75	64.91	69.01
3-ALRSF/ 13000	42/45/13	31.9	-8388.35	OPT	63.07	66.96	57.68	69.25
3-B&B/ 13000	42/45/13	>32	-8956.39	6.34	54.67	50.00	53.00	50.00

In conclusion, the proposed novel ALRSF exhibits multiple improvements with respect to the best subset selection problem (which is an NP-hard problem) within the context of an exact mathematical solution framework. The ALRSF was able to solve large-scale logistic regression classification problems in an efficient and automated manner while providing higher quality solutions when compared to other best subset selection MIP formulations that are commonly found in the literature.

References

Akinwande, O. M., Dikko, G. H., and Samson, A. (2015). Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. *Open Journal of Statistics*, **5**, 754–767.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *Annals of Statistics*, **44**, 813–852.

Civitelli, E., Lapucci, M., Schoen, F., and Sortino, A. (2021). An effective procedure for feature subset selection in logistic regression based on information criteria. *Computational Optimization and Applications*, **80**, 1–32.

Dedieu, A., Hazimeh, H., and Mazumder, R. (2021). Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *Journal of Machine Learning Research*, **22**, 1–47.

Furnival, G. M. and Wilson, R. W. (2000). Regressions by leaps and bounds. *Technometrics*, **42**, 69–79.

Gatu, C. and Kontoghiorghes, E. J. (2006). Branch-and-bound algorithms for computing the best-subset regression models. *Journal of Computational and Graphical Statistics*, **15**, 139–156.

Hastie, T., Tibshirani, R., and Tibshirani, R. (2020). Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. *Statistical Science*, **35**, 579–592.

Heinze, G., Wallisch, C., and Dunkler, D. (2018). Variable selection – a review and recommendations for the practicing statistician. *Biometric Journal*, **60**, 431–449.

Insolia, L., Kenny, A., Calovi, M., and Chiaromonte, F. (2021). Robust variable selection with optimality guarantees for high-dimensional logistic regression. *Stats*, **4**, 665–681.

Kamiya, S., Miyashiro, R., and Takano, Y. (2019). Feature subset selection for the multinomial logit model via mixed-integer optimization. *In The 22nd International Conference on Artificial Intelligence and Statistic*. PMLR, Okinawa, Japan, 1254–1263.

Kolacek, J. and Rezac, M. (2010). Assessment of scoring models using information value. *In 19th International Conference on Computational Statistics*. Elsevier, Paris, France, 1191–1198.

Kutner, M., Nachtsheim, C., Neter, J., and Li, W. (2005). *Applied Linear Statistical Models*. McGraw-Hill.

Kvesic, L. and Gordana, D. (2012). Risk management and business credit scoring. *In In Proceedings of the ITI 2012 34th International Conference on Information Technology Interface*. IEEE, Cavtat, Croatia, 47–52.

Lund, B. (2017). Logistic model selection with SAS PROC's LOGISTIC, HPLOGISTIC HPGENSELECT. *In 2017 Midwest SAS Users Group Conference*. ACM, St. Louis, MI.

Miller, A. (2002). *Subset Selection in Regression*. CRC Press, Washington DC.

Naganuma, M., Takano, Y., and Miyashiro, R. (2021). Feature subset selection for ordered logit model via tangent-plane-based algorithm. *IEICE Transactions on Information and Systems*, **80**, 1–2.

Natarajan, B. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, **24**, 227–234.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, **259**, 801–817.

Sato, T., Takano, Y., Miyashiro, R., and Yoshise, A. (2016). Feature subset selection for logistic regression via mixed integer optimization. *Computational Statistics and Data Analysis*, **64**, 865–880.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, **45**, 427–437.

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic Emergency Medicine*, **18**, 1099–1104.

Takano, Y. and Miyashiro, R. (2020). Best subset selection via cross-validation criterion. *TOP*, **28**, 475–488.

Taşkin, Z. C. (2010). Benders decomposition. *In Wiley Encyclopedia of Operations Research and Management Science*. Wiley, Hoboken, NJ.

Venter, J. v. (2020). *Variable selection in logistic regression using exact optimisation approaches*. Ph.D. thesis, North-West University.

Zhang, Z., Trevino, V., Hoseini, S. S., Belciug, S., Boopathi, A. M., Zhang, P., Gorunescu, F., Subha, V., and Dai, S. (2018). Variable selection in logistic regression model with genetic algorithm. *Annals of Translational Medicine*, **6**, 45–57.

Manuscript received 2022-07-27, revised 2023-03-16, accepted 2023-04-17.