South African Statistical Journal
Vol. 54, No. 1, 93–129
https://doi.org/10.37920/sasj.2020.54.1.6

VARIABLE SELECTION IN LOGISTIC REGRESSION MODELS THROUGH THE APPLICATION OF EXACT MATHEMATICAL PROGRAMMING

J. V. Venter[1]
Centre for Business Mathematics & Informatics, North-West University, Potchefstroom, South Africa
e-mail: *jacques.awe.venter@gmail.com*

S. E. Terblanche
School of Industrial Engineering, North-West University, Potchefstroom, South Africa
e-mail: *fanie.terblanche@nwu.ac.za*

A linearised approximation of the log-likelihood objective function is presented as a potential alternative to iterative fitting methods employed by logistic regression. The log-likelihood objective function is solved using linear programming and a modified version of the linearised logistic regression model is presented, which facilitates best subset variable selection. The resulting model is a mixed integer linear programming problem which incorporates a cardinality constraint on the number of variables. The suggested approach maintains many attractive properties, such as its ability to quantify the quality of the resulting variable selection solution, its independence of the subjective choice of p-values inherent to typical stepwise variable selection approaches, and its capability to edge closer to optimality within increasingly reduced computing times when the correct settings are applied, even for large input datasets.

Computational results are presented to demonstrate the advantages of employing an exact mathematical programming approach towards variable selection in logistic regression applications.

Key words: Best subset selection, Linearisation, Logistic regression, Mixed integer linear programming.

1. Introduction

Logistic regression modelling has been and still remains one of the most frequently applied techniques in a variety of application domains. More specifically, it has gained considerable popularity in the finance industry for its ease of implementation and interpretability (even by stakeholders that are not mathematically inclined), and providing the ability to easily monitor the stability of predictor variables over time (a procedure that is paramount in scorecard modelling).

Although computability is not a concern when fitting logistic regression models, the construction of parsimonious models through the application of variable selection techniques, for instance best subset selection, does pose some challenges. In spite of advances in computing technology over the last decade, practical experience has shown that best subset selection remains an extremely resource

[1] Corresponding author.

MSC2010 subject classifications. 62J12, 90C05, 90C11.

intensive variable selection method, even for a modest number of predictors. Potts and Patetta (1999) explain that best subset selection becomes computationally infeasible when considering data sets comprising more than approximately 40 to 50 inputs. Lund (2017) states that analysts who make use of the software package SAS should consider abandoning best subset selection when the number of variables is 75 or more, citing the exponential increase in execution time. Specifically, best subset selection is considered to be an NP-hard problem[2] (Natarajan, 1995). Most NP problems are notoriously difficult to solve, even by modern software packages. The application of forward and backward selection approaches are practical alternatives, however, it may be to the detriment of optimality.

In this paper, the use of exact mathematical modelling approaches, e.g. mixed integer linear programming (MILP), is suggested as an alternative to traditional stepwise approaches. Solving large scale and/or real-world problems using exact approaches have been neglected for a considerable amount of time within the statistical community due to a widespread belief that such methods may be intractable (Bertsimas and King, 2016). However, significant improvements in computing power and algorithmic advances over the last three decades have resulted in an incredible 200 billion factor speedup in the solving of hard optimisation problems – see e.g. Bertsimas, King and Mazumder (2016).

The exact approach suggested in this paper towards the maximisation of the objective function in logistic regression modelling involves the linearisation of the log-likelihood function such that the posterior probabilities of the model, $P(Y_i = 1|\mathbf{X}_i = x) = \pi_i$, are obtained. More specifically, the maximum likelihood function of the logistic regression problem is formulated as a linear programming problem (LP) and is solved by employing the well-known simplex method, yielding models that are comparable with those produced by standard numerical approaches. In a number of cases, marginally better regression estimates are obtained by the linearised model, as shown in empirical studies in Section 5.

Finally, variable selection is introduced and it is observed that the exact methodology produces accurate and predictive models when concepts of best subset selection are incorporated into the linearised model formulation. The novelty of the suggested approach is the ability to produce proven, optimal and parsimonious logistic regression models. Additionally, empirical evidence suggests that the linearised logistic regression model with best subset selection has the potential to yield solutions that are increasingly closer to optimality within progessively shorter computing times – even for noticably large datasets – when appropriate parameters are used for the MILP formulation.

The rest of this paper is structured as follows: in Section 2 a background on logistic regression, the standard numerical approach used to fit logistic regression models to data and similar work carried out by previous authors, is discussed. In Section 3 a linearised alternative of the log-likelihood function is introduced, followed by a MILP formulation which facilitates variable selection as part of logistic regression. The suggested MILP formulation is also compared with existing methodologies in Section 4. Section 5 encompasses computational results obtained from fitting models to both simulated and real-world data using the methods discussed in Sections 2 and 3 and the subsequent comparison thereof. In Section 6 experimental evidence is presented which allows the reader to analyse the

[2] NP problems are mathematical problems where it is theoretically possible to check the correctness of a solution in polynomial time, but it is very difficult to arrive at the solution itself. NP-hard problems are problems that are believed to be at least as hard as the hardest problems in NP.

trade-off between the quality of solutions obtained and model execution time given different levels of granularity imposed on the linearised approximation of the log-likelihood function. Lastly, in Section 7 a set of summary remarks and a brief discussion around future work conclude the paper.

2. Background and related work

While the subject of variable selection within regression models is vast and encompasses a massive overabundance of approaches, best subset selection will be specifically discussed within the context of the linearised logistic regression formulation. In most applications, the design matrix **X** will consist of a very large number of columns or input variables and it might be unreasonable to expect a classifier to make predictions based on all of these inputs. Ultimately, it would be desirable for any approach to choose the most relevant subset of predictors from the input space that would result in a parsimonious model which can easily be interpreted, but at the same time also produces satisfactory and accurate results. In order to conduct meaningful statistical inference, it might therefore be assumed that the vector of true regression coefficients β may be approximated by a sparse vector that contains many zeros. The best subset selection problem, which chooses a subset of q variables out of the possible p predictors, can be formulated as the following optimisation problem in linear regression where $Y \in \mathbb{R}$ (Miller, 2002):

$$\min_{\beta} \; \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\beta\|_2^2,$$

subject to

$$\|\beta\|_0 \leq q,$$

where $\|\cdot\|_0$ is the L_0-norm and counts the number of non-zeros in β, or $\|\beta\|_0 = \sum_{l=1}^{p} I(\beta_l \neq 0)$ and $I(\,\cdot\,)$ is the indicator function.

In logistic regression, the constraint imposed on the vector of regression parameters in terms of the L_0-norm will remain the same, with only the objective function that changes. In the case where $Y \in \{0, 1\}$, the best subset selection problem becomes

$$\max_{\beta} \quad \log L(\beta),$$

subject to

$$\|\beta\|_0 \leq q.$$

Bertsimas et al. (2016) address the best subset problem via a modern optimisation lens by developing a mixed integer nonlinear optimisation approach (MIO) for choosing q out of p input variables in a linear regression model with n observations and a continuous dependent variable. Additionally, a discrete extension of modern first order continuous optimisation methods that produce high quality feasible solutions is also obtained and used by the authors as warm starts in their MIO. Their resulting approach provides a guarantee on the optimality of the solution, can accommodate side constraints on the regression coefficients, and can be extended to problems with the least absolute deviation loss as objective function (median regression). The authors note that their formulation provides optimal solutions for problems with n in the 1000s and p in the 100s and near-optimal solutions for problems with n in the 100s and p in the 1000s. Both versions of solutions are said to be achieved in minutes. In Bertsimas and King (2016), the work performed in Bertsimas et al. (2016) is extended to produce much more comprehensive regression models. These include conditional constraints that cater for multicollinearity present in the data, variable transformations, forcing certain variables into the model (based on prior knowledge), introducing selective sparsity for variables with a group sparsity structure, and ensuring that no more than one variable from a subset of m variables, where $m < p$, is included in the model. Maldonado, Perez, Weber and Labbe (2014) direct their efforts towards support vector machines (SVMs), as opposed to traditional regression models where the output vector $\mathbf{Y}$ is continuous, and suggest a model formulation similar to that of Bertsimas et al. (2016).

Consider the following MIO formulation from Bertsimas et al. (2016) which addresses best subset selection in linear regression:

$$\min_{\beta} \; \frac{1}{2}||\mathbf{Y} - \mathbf{X}\beta||_2^2, \tag{1}$$

subject to

$$-M_{lL}z_l \leq \beta_l \leq M_{lU}z_l \quad \text{for } l = 1, \ldots, p, \tag{2}$$

$$z_l \in \{0, 1\} \quad \text{for } l = 1, \ldots, p, \tag{3}$$

$$\sum_{l=1}^{p} z_l \leq q. \tag{4}$$

While the objective function itself is of little concern within the context of logistic regression, the constraints that are used to enforce variable selection through mixed integer programming are noted. In the above set of constraints, z_l is a binary decision variable corresponding to the l-th regression coefficient β_l. Clearly, if $z_l = 1$ then β_l is non-zero and included in the model. Alternatively, when $z_l = 0$, the first constraint ensures that β_l is omitted from the final solution. The third constraint subsequently ensures that no more than q variables can be chosen as inputs in the final regression model by forcing the remaining $p - q$ regression coefficients to be exactly equal to zero. The values M_{lL} and M_{lU} serve as lower and upper bounds for the size of the l-th regression coefficient. While the choice of M_{lL} and M_{lU} is subjective, Bertsimas et al. (2016) list a variety of approaches that assist in finding suitable values for these bounds. For numerical stability one would like to choose M_{lL} and M_{lU} such that the differences $-M_{lL} + \beta_l^*$ and $M_{lU} - \beta_l^*$ are as small as possible, where β_l^* is the optimum value of the l-th regression coefficient.

The class of nonlinear regression methods considered in this paper is limited to problems where the vector of dependent variable entries $\mathbf{Y}$ is dichotomous and binary, i.e. each target value Y_i takes a value in a discrete set G that contains two classes, which can be expressed as $Y_i \in G$, where $G = \{0, 1\}$. Obtaining the posterior probabilities $P(Y_i = 1|\mathbf{X}_i = x)$ is usually of great interest within these problem settings. A common practice is therefore the use of logistic regression or probit regression models, as these models produce the desired results by providing the modeller with a posterior probability as output while also maintaining various attractive properties – see Hastie, Tibshirani and Friedman (2001).

From this point forward, *logistic regression* with a binary response vector $\mathbf{Y}$ will be the sole focus in this paper. Kutner, Nachtsheim, Neter and Li (2005) provide a thorough and easily interpretable explanation for the derivation of both logistic regression and probit regression models in the binary case.

A logistic regression model is fitted to the data using the method of maximum likelihood. The resulting fit will produce parameter estimates for the regression coefficients in the logistic response function. When logistic regression models are being considered, it is assumed that the entries of $\mathbf{Y}$ follow an independent Bernoulli dstribution and that each observation has a binary outcome value, or $Y_i \in \{0, 1\}$.

Assuming there are n cases, taking the logarithm of the joint distribution function $L(Y_1, \ldots, Y_n)$ yields the log-likelihood function given by

$$\log L(Y_1, \ldots, Y_n) = \sum_{i=1}^{n} Y_i(\mathbf{X}_i\beta) - \sum_{i=1}^{n} \log[1 + \exp(\mathbf{X}_i\beta)], \tag{5}$$

where Y_i is the response value of the i-th observation, for $i = 1, \ldots, n$, $\mathbf{X}$ is an $n \times p$ design matrix that contains the input the variables, and β is a $p \times 1$ vector of regression coefficients.

By maximising (5), the solution values for the paramter vector β, namely $\hat{\beta}$, are obtained. One of the most popular methods for estimating regression coefficients in logistic regression makes use of a heuristic version of the numerical Newton–Raphson method (Okeh and Oyeka, 2013). Many software packages employ this method in one way or another. When using the Newton–Raphson method to maximise the likelihood, the model starts off with an initial "guess" for the vector of parameter values, after which it subsequently progresses in an iterative fashion. In each step the function is updated by evaluating it at the estimated vector $\hat{\beta}$ obtained from the previous step until the difference in the log-likelihood between the current iteration and previous iteration is insignificant, or $L(\hat{\beta}^{j+1}) - L(\hat{\beta}^{j}) = \epsilon$ for some small value ϵ, where j represents the j-th iteration.

After noting that most commercial MIO software packages struggle to handle nonlinear objective functions effectively (as is the case with logistic regression with feature selection), Sato, Takano, Miyashiro and Yoshise (2015) suggest a piecewise linear approximation of the logistic regression model objective function, which allows the model to be formulated as an LP. Variable selection is then subsequently facilitated by the introduction of integer choice variables to the model constraints (the use of integer choice variables will be discussed in more detail in Section 3). Kamiya, Miyashiro and Takano (2019) embrace the concepts put forth by Sato et al. (2015) – which involve the formulating of dichotomous regression models as MIO problems – by presenting a multinomial logistic regression model with best subset selection as a mixed integer optimisation problem. Both sets of authors indicate that their respective models exhibit favourable generalisation characteristics and are solved within a reasonable amount of time, along with results that rival L_1-regularisation, or lasso (Tibshirani, 1996), and stepwise regression methods.

Sato et al. (2015) first present the logistic regression model as a nonlinear optimisation problem[3]:

$$\min_{\beta} \; 2\sum_{i=1}^{n} f(Y_i(\mathbf{X}_i\beta)), \tag{6}$$

where

$$f(v) = \log(1 + \exp(-v)) \tag{7}$$

is the logistic loss function, which is obtained from the log-likelihood function. As $-\sum_{i=1}^{n} f(Y_i(\mathbf{X}_i\beta))$ is equal to the log-likelihood, the authors choose to minimise $\sum_{i=1}^{n} f(Y_i(\mathbf{X}_i\beta))$ instead of maximising

[3] Note that Sato et al. (2015) make use of $Y \in \{-1, 1\}$ and not $Y \in \{0, 1\}$.

the negative logistic loss function, which produces the same result. This would explain why the log-likelihood in (5) is maximised whereas the objective function in (6) is minimised.

The proposed linearisation approach finds the minimum of the objective function in (6) by a series of linear functions which "cut" away the non-feasible region located below the logistic loss function line. Specifically, given a set of k symmetric grid values $g_j, j = 1, \dots, k$, which are not necessarily equally spaced and where the distance $(g_{j+1} - g_j)$ is not necessarily equal, the function in (7) may be approximated by the pointwise maximum of a set of linear inequalities, or

$$\begin{aligned} f(v) &\approx \max\{f'(g_j)(v - g_j) + f(g_j)\} \quad \text{for } j = 1, \dots, k \\ &= \min\{t | t \geq f'(g_j)(v - g_j) + f(g_j)\} \quad \text{for } j = 1, \dots, k. \end{aligned}$$

The nonlinear model with objective function in (6) is then formulated as the following LP:

$$\min_{\beta} \; 2\sum_{i=1}^{n} t_i, \tag{8}$$

subject to

$$t_i \geq f'(g_j)(Y_i(\mathbf{X}_i\beta) - g_j) + f(g_j) \quad \text{for } j = 1, \dots, k; \; i = 1, \dots, n. \tag{9}$$

Figure 1 illustrates graphically how the objective function is approximated with the set of inequalities in (9) . The solid line represents the logistic loss function evaluated by the nonlinear model shown in (6), while the dashed lines represent a linear approximation of the same function. When the number of grid values k is relatively small, as in Figure 1, a greater disparity exists between the objective function of the nonlinear model and its linearised counterpart. This difference is demonstrated by the shaded triangular region in the figure. The objective function of the piecewise linear model is optimised at the intersection of the two tangent lines, indicated by the solid coordinate. When more linear inequalities are added to the model, the relative gap between the results achieved by the nonlinear model and the approximated linear regression formulation narrows.

In order to accommodate variable selection, Sato et al. (2015) proposed the following mixed integer linear programming model:

$$\min_{\beta, z} \; 2\sum_{i=1}^{n} t_i + F(\sum_{l=1}^{p} z_l + 1), \tag{10}$$

subject to

$$t_i \geq f'(g_j)(Y_i(\mathbf{X}_i\beta) - g_j) + f(g_j) \quad \text{for } j = 1, \dots, k; \; i = 1, \dots, n, \tag{11}$$

$$z_l = 0 \Rightarrow \beta_l = 0 \quad \text{for } l = 1, \dots, p, \tag{12}$$

$$z_l \in \{0, 1\} \quad \text{for } l = 1, \dots, p, \tag{13}$$

where F is a penalty that limits the number of features included in the final model. When $F = 2$, the most optimal model is selected based on the AIC or Akaike Information Criterion (Akaike, 1973), whereas $F = \log(n)$ will result in a final model that was selected using the BIC or Bayesian Information Criterion (Schwartz, 1978).

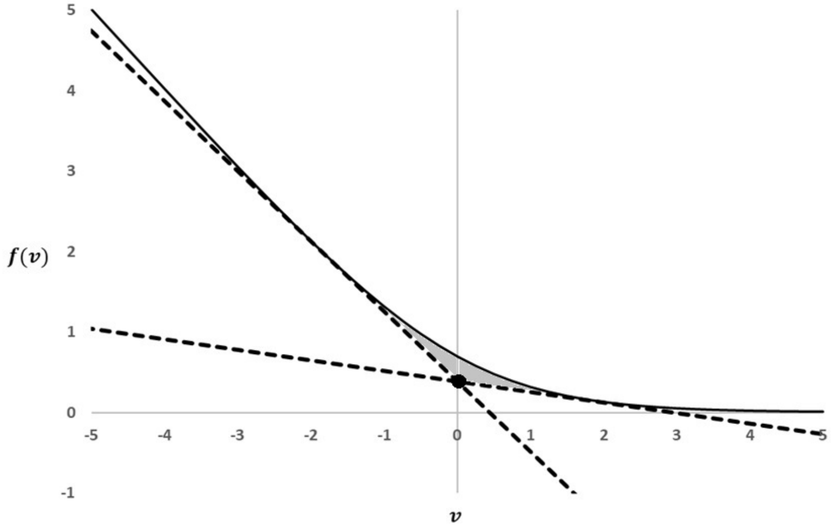

Figure 1. Linearised logistic regression model with two tangent lines.

3. Formulating the log-likelihood function as a MILP within a best subset selection framework

An important attribute of the log-likelihood function is the fact that it is concave. For more than half a century it has been well known that a local optimal solution for a convex function $f(x)$ will also be a global optimal solution. Therefore, a numerical approach taken towards the maximisation of the objective function in (5) would produce a global optimal solution for the estimated coefficient vector $\hat{\beta}$.

The linearisation of the log-likelihood function (5) is facilitated through the use of a grid consisting of k equally spaced grid values g_j, $j = 1, \ldots, k$, spanning the range $[A, B]$, where $A, B \in \mathbb{R}$ and the distance $(g_{j+1} - g_j)$ is identical for all j. For simplification, the choice of A and B used to define the range of the grid can be considered as arbitrary. However, suggestions on selecting a potentially appropriate grid are given in Section 5 of this paper. A linearised version of the log-likelihood objective function is given by the following linear programming problem:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{k} \left(g_j Y_i - \log[1 + \exp(g_j)]\right) \lambda_{ij}, \tag{14}$$

subject to

$$\sum_{j=1}^{k} g_j \lambda_{ij} = \mathbf{X}_i \beta \quad \text{for } i = 1, \ldots, n, \tag{15}$$

$$\sum_{j=1}^{k} \lambda_{ij} = 1 \quad \text{for } i = 1, \ldots, n, \tag{16}$$

$$0 \leq \lambda_{ij} \leq 1 \quad \text{for } j = 1, \dots, k;\ i = 1, \dots, n, \tag{17}$$

where k is the number of grid values specified in the linearisation, g_j is the j-th grid value, and λ_{ij} is the j-th weight value associated with the j-th grid value for the i-th observation.

Theorem 1. For a small enough grid interval $(g_{j+1} - g_j)$, the linear programming problem (14)–(17) provides an optimal solution to the log-likelihood function given by

$$\log L(Y_1, \dots, Y_n) = \sum_{i=1}^{n} Y_i(\mathbf{X}_i\beta) - \sum_{i=1}^{n} \log[1 + \exp(\mathbf{X}_i\beta)].$$

Proof. For $k = 2$, we have from constraint (15) that $\lambda_{i1}g_1 + \lambda_{i2}g_2 = \mathbf{X}_i\beta$, for $i = 1, \dots, n$. For a specific case i, it can be shown that

$$\begin{aligned} Y_i(\mathbf{X}_i\beta) - \log[1 + \exp(\mathbf{X}_i\beta)] &= Y_i(\lambda_{i1}g_1 + \lambda_{i2}g_2) - \log[1 + \exp(\lambda_{i1}g_1 + \lambda_{i2}g_2)] \\ &\leq Y_i(\lambda_{i1}g_1 + \lambda_{i2}g_2) - (\lambda_{i1}\log[1 + \exp(g_1)] + \lambda_{i2}\log[1 + \exp(g_2)]), \end{aligned}$$

since $\lambda_{i1}g_1 + \lambda_{i2}g_2$ is a convex combination of the adjacent grid points g_1 and g_2, according to constraints (16) and (17). Therefore, for a large enough k, $g_{j+1} \approx g_j$ so that

$$\begin{aligned} Y_i(\mathbf{X}_i\beta) - \log[1 + \exp(\mathbf{X}_i\beta)] &\approx Y_i\left(\sum_{j=1}^{k} \lambda_{ij}g_j\right) - \sum_{j=1}^{k} \lambda_{ij}\log[1 + \exp(g_j)] \\ &= \sum_{j=1}^{k} \left(g_jY_i - \log[1 + \exp(g_j)]\right)\lambda_{ij}. \qquad \blacksquare \end{aligned}$$

Similar to the approach by Bertsimas et al. (2016), variable selection for logistic regression is introduced by reformulating (14)–(17) as a MILP. More specifically, the binary decision variable z_l is introduced to determine whether the l-th variable is included in the model or not. If $z_l = 1$ then β_l is non-zero and included in the model. Alternatively, when $z_l = 0$, β_l is omitted from the final solution. The parameter q is introduced to limit the number of variables selected for inclusion in the final model and the values M_{lL} and M_{lU} serve as lower and upper bounds for the size of the l-th regression coefficient.

The objective of the logistic regression problem with variable selection is to

$$\max \sum_{i=1}^{n}\sum_{j=1}^{k} \left(g_jY_i - \log[1 + \exp(g_j)]\right)\lambda_{ij}, \tag{18}$$

subject to

$$\sum_{j=1}^{k} \lambda_{ij}g_j = \mathbf{X}_i\beta \quad \text{for } i = 1, \dots, n, \tag{19}$$

$$\sum_{j=1}^{k} \lambda_{ij} = 1 \quad \text{for } i = 1, \dots, n, \tag{20}$$

$$0 \leq \lambda_{ij} \leq 1 \quad \text{for } j = 1, \dots, k;\ i = 1, \dots, n, \tag{21}$$

$$-M_{lL} z_l \leq \beta_l \leq M_{lU} z_l \quad \text{for } l = 1, \ldots, p, \tag{22}$$

$$z_l \in \{0, 1\} \quad \text{for } l = 1, \ldots, p, \tag{23}$$

$$\sum_{l=1}^{p} z_l \leq q. \tag{24}$$

4. Comparison with existing approaches

The formulation of the linearised logistic regression model with best subset selection shown in (18)–(24) is premised on the same motivations listed by Sato et al. (2015), which mainly involve three critical concepts, namely:

- As is the case with the model presented by Bertsimas et al. (2016), Sato et al. (2015) make use of binary z-variables to facilitate variable selection, which has proven to be useful when feature selection is carried out within a mixed integer optimisation framework.
- A linear approximation of the nonlinear logistic regression model objective function is proposed so as to allow the regression model with variable selection to be formulated as a MILP.
- The use of a mixed integer linear optimisation model is suggested so as to provide the end user with a guarantee on the optimality of the solution, as opposed to more traditional variable selection methods that depend on the evaluation of p-values and variable significance levels[4].

Indeed, the model shown in (18)–(24) and MILP proposed by Sato et al. (2015) share many similarities, especially considering that each observation, for $i = 1, \ldots, n$, is associated with k constraints in the constraint matrix. However, a key difference that exists between the two MILP formulations is the fact that the logistic regression model in (10)–(13) attempts to linearise the logistic loss function, whereas the model suggested in (18)–(24) attempts to directly linearise the log-likelihood function itself. Consider Figure 2 and the subsequent discussion thereof.

In Figure 2 the nonlinear log-likelihood function, which is equal to $-f(v)$ in (7), is plotted as a solid line. The two dashed lines represent the linear approximation of the logistic loss function as put forth by Sato et al. (2015) in (8)–(9), which was also shown previously in Figure 1. The dotted lines visually represent the linearisation of the log-likelihood function proposed in (18)–(24). If the number of grid values k is not substantially large, a larger gap will exist between the optimal objective function value achieved by both linearised models and the maximum log-likelihood value found by the nonlinear model. Specifically, the model in (10)–(13) will be optimised somewhere close to coordinate P_1, whereas the logistic regression model in (18)–(24) will yield an optimal log-likelihood value that is located on the dotted line. In fact, the maximised log-likelihood value will be a convex combination of two adjacent grid values (note that the grid values are shown as empty circle coordinates in Figure 2).

The following notable differences exist between the two linearised logistic regression models:

[4] Kutner et al. (2005) state that stepwise regression approaches, such as forward and backward regression, will produce a single "good" model fit. However, these methods fail to provide the modeller with some sort of a guarantee which shows that the resulting solution is the best possible model found amongst all combinations of predictors.

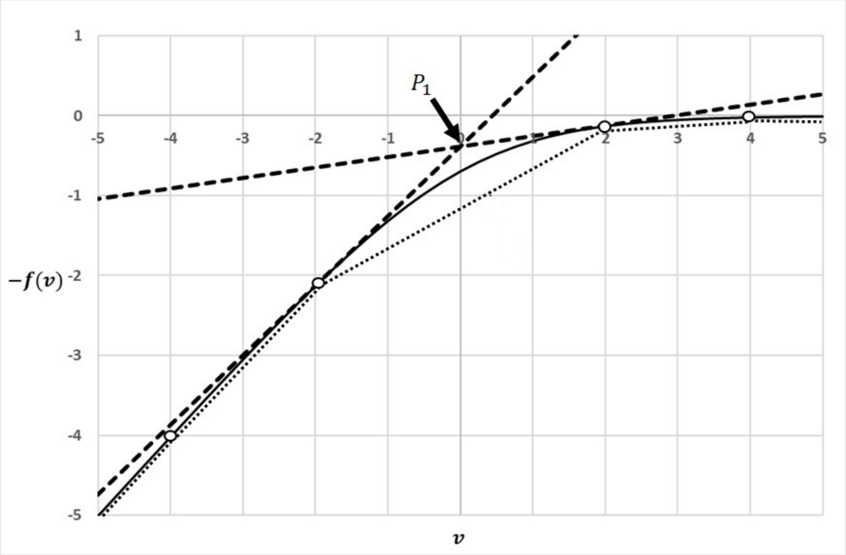

Figure 2. Comparison of two linearised logistic regression models.

1. Let k_{max} be a significantly large number that tends to infinity. While $k \ll k_{max}$, the linearised model in (10)–(13) will tend to overestimate the log-likelihood function, whereas the linearised model in (18)–(24) will tend to underestimate it. Simply put, the proposed model in (18)–(24) yields a more conservative estimate and provides assurance that model fit can only be improved.

2. Sato et al. (2015) do not explicitly specify a cardinality parameter to facilitate variable selection, as is the case with constraint (24) in the model in (18)–(24) and constraint (4) in the model proposed by Bertsimas et al. (2016). Instead, the authors include a penalty term in the objective function, similar to the Lagrangian form of lasso. In turn, the penalty term performs feature selection based on the AIC or BIC. Empirical studies have shown that criteria such as the AIC or BIC have a tendency to favour models with an overabundance of predictors, unless the sample size n is sufficiently large (Kutner et al., 2005).

3. A different approach is followed for selecting the appropriate grid values g_j, $j = 1, \ldots, k$, that are utilised in the linearisation performed by the two respective models. Specifically, Sato et al. (2015) propose a greedy algorithm that sequentially adds tangent lines one by one to the model whereby each additional linear inequality attemps to reduce the area of the shaded triangle shown in Figure 1, thereby decreasing the gap that exists between the nonlinear model and its linearised version. While this method is quick and effective, it produces a grid of values that are not particularly intuitive if they need to be specified explicitly – especially for a novice modeller. Alternatively, the suggested model in (18)–(24) does not add grid values iteratively, but instead requires the modeller to specify the beginning and end points of the grid, along with the number of grid values k that the user would like the model to employ during its execution. It will then proceed to utilise a grid of k equally spaced knots that lie within the

aforementioned specified range. In Section 5.6 of this paper, an approach is suggested that provides the user with a relatively easy-to-follow and less mathematically intensive method for determining a suitable grid of coordinates. Ultimately, the modeller will have to determine his or her preference towards algorithmic transparency and ease of understanding versus his or her inclination towards model autonomy and speed of execution when considering the two approaches.

5. Results

In order to assess the performance of the proposed formulations presented in equations (14)–(17) and (18)–(24), the linearised logistic regression model with variable selection was implemented and solved using the commercial product IBM ILOG CPLEX Optimization Studio, version 12.6. The results of these efforts were then compared to runs that were executed on the same datasets using SAS's PROC LOGISTIC procedure in a 64-bit version of SAS Enterprise Guide 7.1. Note, however, that IBM ILOG CPLEX is a mathematical programming solver and not a statistical analysis tool.

Ultimately, obtaining results comparable to (or better than) those achieved in SAS would be ideal, since its PROC LOGISTIC procedure is considered to be the benchmark in this exercise. Additionally, all simulated datasets that were used for synthetic model runs were generated in SAS Enterprise Guide. All work was carried out on a desktop computer with Intel i7-3770 3.40GHz processor, 16 GB of RAM and 64-bit Windows 7 operating system.

The reader should note that, throughout the section, the term "CPLEX run" refers to the linearised formulation produced in equations (14)–(17) and (18)–(24), whereas the term "SAS run" refers to the same model carried out using SAS's logistic procedure. Lastly, it should be known that for Sections 5 and 6, the quantities p and q *do not* include the intercept term of the logistic regression model. This means that a logistic regression problem that considers a total of p potential features will be based on a design matrix with $p + 1$ columns, where an additional column of 1s is added for the intercept term.

5.1 Simulated data runs

To perform comparisons on simulated data, a design matrix $\mathbf{X}$ that contains p input variables from a standard normal distribution for a total of n observations, was generated. To obtain a binary response vector $\mathbf{Y}$, the following steps were carried out:

1. Generate an $n \times 1$ vector $\mathbf{U}$ that contains n uniformly distributed random variables.

2. Generate an $n \times 1$ vector $\mathbf{L}$ that contains n random variables having a logistic distribution by utilising the vector $\mathbf{U}$ obtained in step 1. The i-th entry of $\mathbf{L}$ is given by $L_i = \log[U_i/(1 - U_i)]$.

3. Generate an $n \times 1$ vector $\mathbf{Y}^c$ which represents a latent output variable with a logistic error distribution, where the i-th entry of $\mathbf{Y}^c$ is given by $Y_i^c = \sum_{l \in Q} \beta_l^* X_{il} + L_i$. In the aforementioned summation, $\mathbf{X}$ is the input variable matrix containing p variables from a standard normal distribution, $\beta_l^* = 1$ for $l \in Q$ and $\beta_j^* = 0$ for $j \notin Q$. β_l^* is the true underlying value specified for the l-th regression parameter.

4. To produce a binary response vector containing either 0 or 1 entries, an $n \times 1$ vector $\mathbf{Y}$ was generated, where $Y_i = 1$ if $Y_i^c \geq 0$, else $Y_i = 0$ if $Y_i^c < 0$.

Initially, a sample containing $n = 1\ 000$ observations with $p = 100$ standard normal random variables was generated. A column of 1s was concatenated horizontally to the matrix of input variables in order to compensate for the intercept of the regression model, meaning that the design matrix contains a total of $p+1$ columns. A set of 10 equi-spaced variables was then chosen to influence the response and has true regression coefficients greater than zero. Given that X_0 represents the intercept term, the variables $X_1, X_{11}, X_{21}, X_{31}, X_{41}, X_{51}, X_{61}, X_{71}, X_{81}$ and X_{91} were all selected to be associated with a regression coefficient of one, i.e. $\beta_l^* = 1$ where $l \in Q$ and Q is the set $\{1, 11, 21, 31, 41, 51, 61, 71, 81, 91\}$. The response vector was then constructed as a linear combination of these 10 variables using steps 3 and 4 outlined above. The dataset contained a total of 500 "goods" or observations with $Y = 0$ and 500 "bads" or observations with $Y = 1$. Note that no variable selection was performed in either SAS or CPLEX during this simulation. The purpose is to first establish whether the proposed linearised formulation in (14)–(17) yields the desired results when used to fit a logistic regression model before cardinality constraints come into play.

The logistic regression model in SAS achieves a maximum log-likelihood of -273.836. The iterative algorithm employed by SAS appears to be quite successful in separating the estimated regression coefficients associated with the 10 variables mentioned above (that were set to influence the target variable) from the parameter estimates obtained for the other 90 variables. The fitted coefficient values for the entire collection of non-influential predictors are scattered around the zero-line, with the largest deviations being close to 0.2 or -0.2. Alternatively, all parameter estimates that were found for the 10 chosen variables had the correct sign and were far removed from the other fitted coefficients, with most estimates scattered between 1.4 and 1.5 (with the exception of the coefficient obtained for X_{11}). Overall, the results can be interpreted as desirable, given the clear separation between the parameter estimates for the two groups and their relative close proximity to their true underlying values. The fact that the model tends to overestimate the coefficients for the 10 variables that were selected to influence the response, is noted, with a difference of roughly 0.5 between the estimated and true underlying parameter values for the majority of these inputs. However, given the relatively small size of the sample, this can be expected to improve with an increase in the number of observations.

Using a grid of 401 knots that span between -20 and 20 with 0.1 intervals, a logistic regression model is fitted to the data using the linearised model in (14)–(17). The model produces identical results to those found in the SAS run, with a maximum log-likelihood of -273.9 achieved in 20.56 seconds, along with parameter estimates that are nearly indistinguishable from those obtained by SAS's iterative algorithm. Consequently, the same interpretation applies.

A second run was performed wherein the range of the grid used in the linearisation was kept the same (-20 and 20), but intervals were specified to have a length of 0.01, resulting in a total of 4 001 grid values (as opposed to the 401 knots used in the previous run). This was done in order to ascertain whether or not an improvement in the model can been seen by reducing the distance between the grid values (this is based on the notion – as shown in Section 3 – that as the distance between the knots tends to zero, the continuous log-likelihood function employed by the iterative algorithm is obtained). While the linearised model only produces results after 3 minutes and 24 seconds (roughly 3 minutes more than needed by the previous run where only 401 grid values are employed), barely no real increase in the log-likelihood is recorded, with a maximum objective function value of -273.836 obtained (an improvement of only 0.044 over the previous run). The estimated parameter estimates

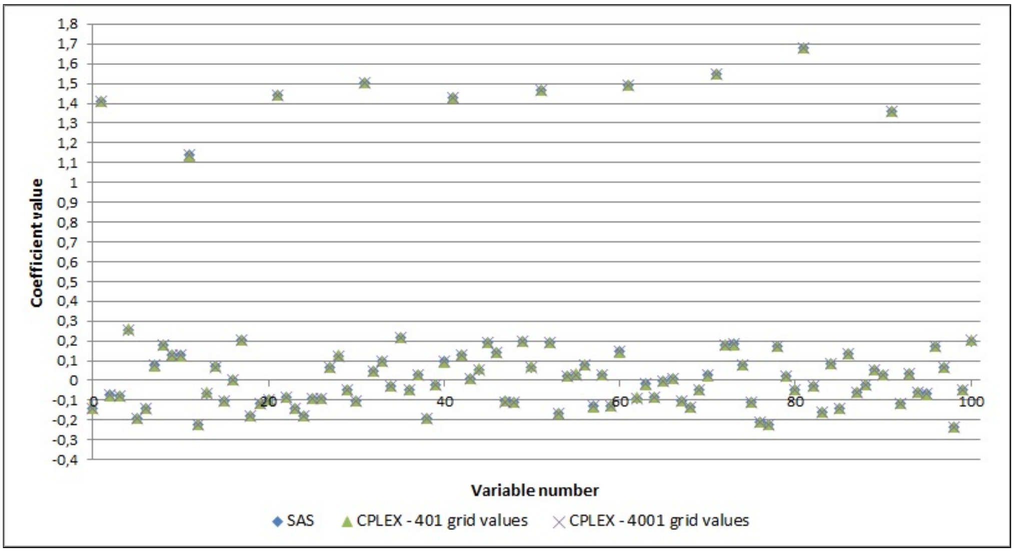

Figure 3. Regression parameter estimates obtained by SAS and CPLEX model runs.

Table 1. Results of CPLEX and SAS runs.

n	$\log L$**: CPLEX linearised run**	$\log L$**: SAS logistic procedure**
3 000	−957.671	−957.427
5 000	−1 704.744	−1 704.3
10 000	−3 447.104	−3 446.2
20 000	−7 001.821	−6 999.979
50 000	−17 596.363	−17 577.649

were nearly identical to those fitted during the previous run. The empirical studies presented in Section 6 examine the trade-off between model execution time and the accuracy of results when the number of grid values which are employed in the linearised approximation, is varied.

Figure 3 illustrates the regression coefficient estimates produced by the aforementioned SAS and CPLEX runs. The x-axis denotes the variable number, from 0 to 100, while the y-axis represents the corresponding value of the estimated regression parameter for the variable in question. Note how the fitted coefficients produced for the 10 variables that were selected are located far above the rest and tend to be closer to one, while the rest are situated closer to zero.

Runs were then executed where the number of observations contained in the dataset, n, was increased. Table 1 shows log-likelihood values obtained for different values of n and the subsequent comparison thereof with the maximum likelihood estimate (MLE) produced by SAS.

While not shown graphically in this paper, it should be noted that as the number of observations, n, in the training set increases, both the linearised model used in CPLEX and the iterative model employed by SAS obtain parameter estimates that are located increasingly closer to their true underlying values. As the number of records within the design matrix on which the classifier is being trained becomes greater, it allows the model to produce a solution for the estimated coefficient vector that is a much less biased estimate of the true underlying parameter vector (Potts and Patetta, 1999). At $n = 20\,000$ observations, the estimated coefficients for the 10 inputs that were chosen to influence the response (with true underlying parameter of $\beta^* = 1$) were close to 1, while the fitted parameters for the remaining 90 variables were all found within an extremely narrow range centered around the zero-line, with all values between -0.05 and 0.05. At $n = 50\,000$, fitted coefficients for the selected 10 variables were almost exactly equal to one, while the parameter estimates for the remaining 90 variables were scattered close to the zero-line. Notice from Table 1, however, that the log-likelihood values produced by the linearised model in CPLEX are marginally lower than the objective function values obtained in SAS. This can be attributed to the fact that the proposed model in (14)–(17) serves as a linear approximation of the log-likelihood function in (5). The model in CPLEX will only yield maximum likelihood function values identical to those in SAS when k is substantially inflated and tends to infinity.

Overall, the proposed linearised model appears to fare well and yields log-likelihood objective function values that compare favourably with the nonlinear model in SAS. The parameter estimates obtained for the variables chosen to influence the response (with true underlying value of one) are forced to be much higher, while the fitted coefficients produced for the set of unselected inputs (with true underlying value of zero) are scattered within an increasingly narrow range about the zero-line. It should be noted, however, that these results are based on a design matrix with a very weak inter-variable correlation structure. The next section explores the behaviour of the proposed linearised model fitted to data with correlated outputs.

5.2 Simulated data runs with correlated inputs

For these simulations, the same approach outlined in Section 5.1 was utilised. However, a key difference to note is that linear correlations were deliberately introduced to create between-variable correlations within the design matrix. Specifically, correlated variables were generated such that the ij-th entry of the correlation matrix is given by $\rho^{|i-j|}$ for some specified correlation value ρ. This was done by using a variation of Cholesky's decomposition (Press, Flannery, Teukolsky and Vetterling, 1992) outlined in the following steps:

1. For some specified value ρ, generate a correlation matrix $\mathbf{\Gamma}$ such that the ij-th entry is given by $\Gamma_{ij} = \rho^{|i-j|}$.
2. Obtain a $p \times p$ diagonal matrix $\mathbf{D}$ where the entries along the diagonal are given by $D_{ii} = \sqrt{\mathbf{\Gamma}_{ii}}$.
3. Find a $p \times p$ matrix $\mathbf{R}$ by setting $\mathbf{R} = \mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1}$.
4. Find a $p \times p$ matrix $\mathbf{C}$ where the entries of $\mathbf{C}$ are obtained as follows:
   a. $C_{ij} = R_{ij}/\sqrt{R_{ji}}$ if $j = 1$,
   b. $C_{ij} = 0$ if $j > i$,
   c. $C_{ij} = \sum_{k=1}^{j-1} \left[C_{ik}C_{jk}\right] / \left(\sum_{k=1}^{j-1} C_{jk}^2\right)$ for $j < i$ and $j \neq i$.
5. Obtain a new design matrix $\mathbf{X}_{new}$ by multiplying the current input variable matrix containing p standard normal random variables with the transpose of $\mathbf{C}$, or $\mathbf{X}_{new} = \mathbf{X}\mathbf{C}^T$.

The new design matrix obtained in step 5 will now have a correlation matrix closely resembling the one specified in step 1.

For the first correlations run, 500 observations and 100 variables from a standard normal distribution were generated. Once more, a column of 1s is added to the matrix of inputs to compensate for the intercept term of the regression model. By using the steps outlined above, correlations are introduced into the design matrix by specifying $\rho = 0.5$ as set out in step 1. Again, 10 equi-spaced variables are chosen to have an effect on the response by stipulating that the inputs $X_1, X_{11}, X_{21}, X_{31}, X_{41}, X_{51}, X_{61}, X_{71}, X_{81}$ and X_{91} all have a true regression coefficient of one, i.e. $\beta_l^* = 1$, $l \in Q$, where β_l^* is the true underlying value of the regression parameter. The binary dependent vector $\mathbf{Y}$ is obtained in the same way as before. The dataset contained a total of 254 "goods" or observations with $Y = 0$ and 246 "bads" or observations with $Y = 1$.

When $\rho = 0.5$, SAS obtains a maximum log-likelihood value of -107.643. The model fitted by the linearised logistic regression executed within CPLEX, on the other hand, attained a log-likelihood value of -108.328 after 10.38 seconds of execution by utilising a grid of 401 knots spanning the range $[-20, 20]$ with intervals of 0.1. Additionally, the linearised model was refitted in CPLEX where a grid consisting of 4 001 knots were used, once again with a range of $[-20, 20]$, but with 0.01 intervals instead. As with previous simulations, this was done in order to gauge whether or not the linearised model produces more accurate results alongside an increase in the number of grid values. However, after fitting the model for 1 minute and 51 seconds, an objective value of -108.305 for the likelihood function was obtained, resulting in a meager increase of 0.023, or 0.02%. In contrast, the time consumed by fitting the model increased nearly ten-fold by 969.36%. After noting that no substantial difference exists between the parameter estimates obtained by the two linearised model runs, it can be inferred – as in the Section 6.1 – that a substantial increase in the size of the grid in the linearised model does not necessarily amount to a meaningful improvement in accuracy.

When the parameter estimates themselves are analysed, it can be seen that very little difference exists between the fitted regression coefficients produced by the two respective models. It is noted that the coefficients obtained by both SAS and CPLEX deviate quite wildly from their true underlying parameter values. However, this can be expected in the presence of moderate to severe linear correlations that exist between input variables. On a positive note, both algorithms appear to be quite successful in creating two distinct groups of estimates by assigning much higher fitted coefficient values to the 10 variables chosen to influence the response, as opposed to the lower estimate values obtained for the other 90 design matrix columns. Ultimately, this will result in the original 10 variables having a much larger influence on the scores obtained for new observations, which is a welcome attribute. Finally, it can be seen that CPLEX produces estimated coefficients that are slightly closer to their true underlying specified values as opposed to those fitted by SAS. The estimates of the 10 chosen variables are vaguely closer to one, while the fitted coefficients associated with the remaining 90 inputs are scattered marginally closer towards the zero-line. Figure 4 provides a graphical representation of estimated coefficients obtained by each model where $\rho = 0.5$.

In order to assess the validity of using (18)–(24) to facilitate variable selection in regression models, both SAS and CPLEX models are refitted to the dataset with $\rho = 0.5$, but a cardinality constraint q was added to the models during the second round of model execution. Since it is known that 10 equally spaced inputs were selected to influence the response, the cardinality parameter was set to $q = 10$,

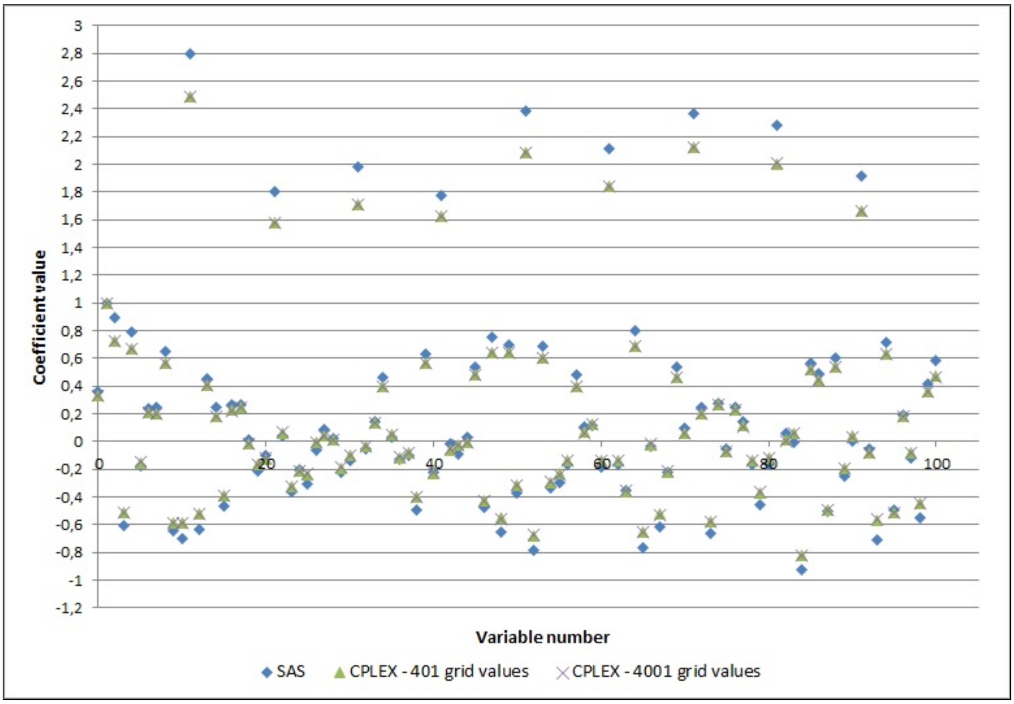

Figure 4. Regression parameter estimates obtained by SAS and CPLEX models when $\rho = 0.5$.

meaning that both models were instructed to have exactly 10 non-zero estimated parameters[5] in their final solution vectors. Best subset selection was performed in SAS by using the SELECTION = SCORE option in the PROC LOGISTIC statement, along with the options START=10 and STOP=10. This allows SAS to only return the best 10-variable model. SAS utilises the *Leaps and Bounds* or LBA method of Furnival and Wilson (1974) to perform best subset selection. SAS attained a 10-variable model with a log-likelihood of -164.698, while the linearised model in CPLEX produced 10 estimates for the exact same variables chosen by the SAS best subset selection procedure, yielding a final likelihood value of -165.243. Both models selected non-zero estimates for the 10 initial variables chosen to have true regression coefficients of one and no significant difference was found between the two solution vectors. Furthermore, both models obtained final coefficient estimates that were very close to their true underlying values, with slight deviations for X_{11} ($\hat{\beta}_{11} = 1.1336$), X_{21} ($\hat{\beta}_{21} = 0.8011$) and X_{71} ($\hat{\beta}_{71} = 1.3081$). Table 2 shows the estimated regression coefficients for the 10 selected variables in both models.

For the second and third correlation exercises, the correlation parameter ρ was specified to be 0.8 and 0.85, respectively, resulting in much more severe linear correlations being present in the training data. SAS displayed log-likelihood values of -78.789 for $\rho = 0.8$ and -89.862 for $\rho = 0.85$, whereas the linearised model concluded with objective function values of -82.181 for $\rho = 0.8$ and -94.473 for $\rho = 0.85$. As can be expected, the parameter estimates obtained by both models were noticeably

[5] The number of non-zero parameter estimates becomes 11 when the intercept is also considered.

Table 2. Estimated regression coefficients obtained by SAS and CPLEX when cardinality constraints are applied and $\rho = 0.5$.

Variable	CPLEX fitted coefficient	SAS fitted coefficient
X_1	0.9475	0.9579
X_{11}	1.1336	1.1360
X_{21}	0.8011	0.8157
X_{31}	1.0249	1.0249
X_{41}	1.0041	1.0119
X_{51}	1.0101	1.0253
X_{61}	0.9924	0.9927
X_{71}	1.3081	1.3076
X_{81}	1.0129	1.0081
X_{91}	0.9999	1.0035

different from their true underlying values that were specified during the creation of the datasets, with the models failing to establish two distinct groups of estimates like those seen previously when the correlation coefficient was not as extreme. This resulted in a final solution vector where the fitted parameters seem to be far removed from their true underlying values for both models. Once more, as in the case where $\rho = 0.5$, it was noted that the linearised model obtains fitted coefficients that are located marginally closer to their true specified values. Figure 5 plots the results for both models when $\rho = 0.8$.

5.3 Tests on real-world data: HEART dataset

In this exercise, the performance of the proposed linearised model is tested on a real-world dataset that is openly available for download. It should be noted that for this test, equation (14)–(17) is tested first, meaning that a linearised model without subset selection is fitted to the data. This is done to gauge whether or not the linearised model produces results comparable to the benchmark when applied to data that were not simulated. Afterwards, the proposed model in (18)–(24) is fitted in order to perform best subset selection on the same set of data.

Consider the HEART dataset that is readily available under the SASHELP library in SAS Enterprise Guide or Base SAS and contains the results of the Framingham Heart Study (SAS Institute Inc., 2017). The dataset comprises of 5 209 observations with various attributes. The response variable in question is called *Status* and is equal to 1 if the subject died and 0 if the subject survived. Of the 5 209 subjects, 3 218 survived ($Y = 0$) and 1991 died ($Y = 1$). Table 3 lists the 12 predictors that exist within the dataset (note that variables marked with an [*] were not supplied within the dataset, but were custom inputs that were created based on the data that were available).

All variables that contained missing values (except for the *Smoking* variable) were imputed with the median value of the dataset for the predictor in question. In the case of the *Smoking* variable, the value was set equal to 0 if it was missing. Finally, using a 50/50 split, half of the dataset was selected at random to train the models. The other half was used to test the generalisation abilities of both models.

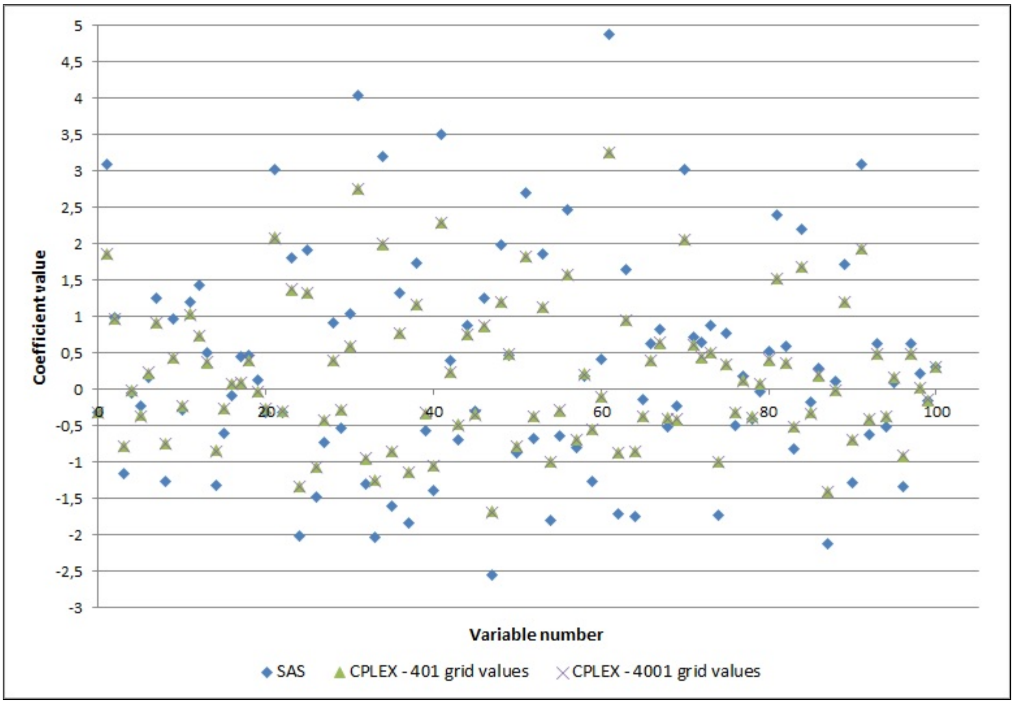

Figure 5. Regression parameter estimates obtained by SAS and CPLEX models when $\rho = 0.8$

The solutions obtained from the linearised approach as well as SAS's PROC LOGISTIC procedure are nearly identical, with virtually no difference between the estimated regression coefficient vectors. In either case, the models choose relatively high coefficient estimates for the subject's age, whether the subject was diagnosed with cardiovascular disease, whether the subject smokes, and the subject's gender, while setting the parameter estimates for the remaining variables close to zero. While the cardinality constraints discussed in equation (18)–(24) were not utilised during this round of model execution, the results suggest that if best subset selection was applied, the estimated coefficients of the remaining variables could have been set exactly equal to zero without much loss of accuracy. Figure 6 illustrates graphically the estimates produced by the linearised model in CPLEX (again, note that SAS's logistic procedure yielded identical solutions).

Since the solutions provided by both approaches are the same, it should come as no surprise that the models obtained from both SAS and CPLEX display similar misclassification rates and Gini coefficients. While the linearised model attains a 24.07% misclassification rate on the training set and 24.62% misclassification rate on the test set, the SAS logistic regression model displays misclassification rates of 24.14% and 24.39% on the training and test sets, respectively. A Gini[6] coefficient of 63.79 obtained from the CPLEX model is slightly higher than the 63.75 Gini value

[6] A Gini coefficient is often used in conjunction with or instead of the c-statistic or area under the curve (AUC), all of which are ranking tools utilised to measure model performance and have similar interpretations (Rezac and Rezac, 2011). Specifically, AUC = (Gini + 100)/2.

Table 3. List of predictor variables from the HEART dataset used as inputs in regression.

Variable name	Description
AgeAtStart	Age of subject
Height	Height of the subject
Weight	Weight of the subject
Diastolic	Diastolic blood pressure reading
Systolic	The subject's systolic blood pressure pressure reading
MRW	The subject's Metropolitan Relative Weight (similar to Body Mass Index)
Smoking	Number of cigarettes smoked by the subject (0 for non-smokers)
Cholesterol	The subject's cholesterol level
CHD_ind*	Indicates whether the subject was diagnosed with cardiovascular heart disease (1 if yes, 0 if no)
Smoke_ind*	Indicates whether the subject smokes or not (1 if yes, 0 if no)
BP*	Indicates whether the subject has high blood pressure or not (1 if yes, 0 if no)
Sex_bin	The gender of the subject (1 if male, 0 if female)

seen for the SAS model when the ranking abilities of the models are evaluated on the training set, while Gini coefficients of 61.29 for the CPLEX model and 61.36 for the SAS model were achieved separately on the test set.

Next, the performance of the linearised model with best subset selection was tested. Since the approach outlined in (18)–(24) was being utilised during the second round of model fitting, it meant that the cardinality constraint q had to be explicitly specified. This was done by executing SAS's version of best subset selection by using the SELECTION = SCORE option in the PROC LOGISITIC model statement and selecting BEST = 1. By specifying these arguments, SAS will fit the best possible model containing l variables, for $l = 1, \dots, p$, where a model with p non-zero estimated regression coefficients represents the full model with all of the input variables added to the final output. In other words, SAS finds the best 1-variable model, after which it finds the best 2-variable model and proceeds through the list of inputs by repeating this exercise until all of the design matrix columns are included in the last model iteration. After each model was fitted, the AUC (area under the curve) was reported. This was done to determine the point at which the performance of the model starts to diminish, meaning that the addition of more variables to the mix does not improve model fit. Figure 7 illustrates that a model containing five input variables should suffice (indicated by the square marker). Notice that the AUC starts to level off beyond this point, implying that all subsequent models would deliver similar predictive performance.

By specifying $q = 5 + 1$, (one more to make provision for the intercept term), the model in (18)–(24) was executed in CPLEX and yielded a maximum log-likelihood of -1305.83, while its counterpart fitted in SAS achieved a near-identical MLE of -1305.47. Both approaches opted to include the same five variables in the final model, along with estimated regression coefficients that are nearly indistinguishable for the selected inputs and the intercept term. Furthermore, accuracy statistics and Gini values were obtained which are strikingly similar to those seen previously when no feature selection was applied. This supports the assertion made earlier which states that excluding the remaining inputs in the design matrix (barring the five variables that were selected) will most likely not have an adverse affect on the model's predictive prowess.

Table 4 displays the estimated regression coefficients obtained by the two respective models when

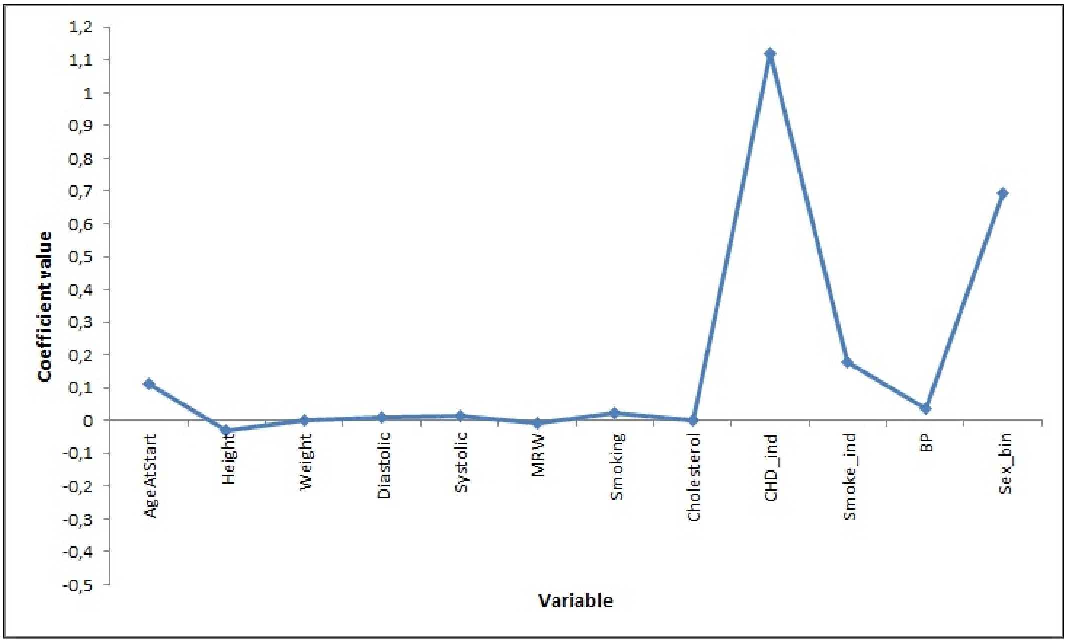

Figure 6. Coefficient estimates obtained by the linearised logistic regression model in CPLEX for the HEART dataset.

best subset selection is applied, while Table 5 contains model evaluation metrics. From Table 5 it can be seen that the linearised model with best subset selection outperformed the model fitted in SAS by a small margin.

5.4 Tests on real-world data: JUNKMAIL dataset

For the second real-word exercise, a dataset within the SASHELP library called JUNKMAIL was used. The data were collected by Hewlett-Packard labs (HP Computers) and subsequently donated by George Forman. The set holds data relating to 4 601 emails that were classified as SPAM ($Y = 1$) or PERSONAL / WORK-RELATED ($Y = 0$). The data are also openly available on the UCI Machine Learning Repository. There are 57 numeric predictor variables, most of which relate to metrics that indicate how frequently a certain word occurs within the email itself. Additionally, the last three variables are considered to be run-length attributes and measure the length and sequences in consecutive capital letters contained within the email. Already provided in the data was an indicator column dubbed *Test*, which splits the data into a model training set and a test set. The training set contains 3 065 observations, of which 1 847 correpsond to $Y = 0$, while 1 218 have a response of $Y = 1$. This particular dataset was chosen due to the fact that it contains a moderate number of observations (4 601) and a relatively large number of columns (57 predictors), which makes it more relatable to and representative of an actual dataset that a user might encounter in daily life.

Exploratory data analysis revealed that most of the numeric variables contained a large number of zeros as entries, with the majority of inputs consisting only 30% or less of non-zero values (with

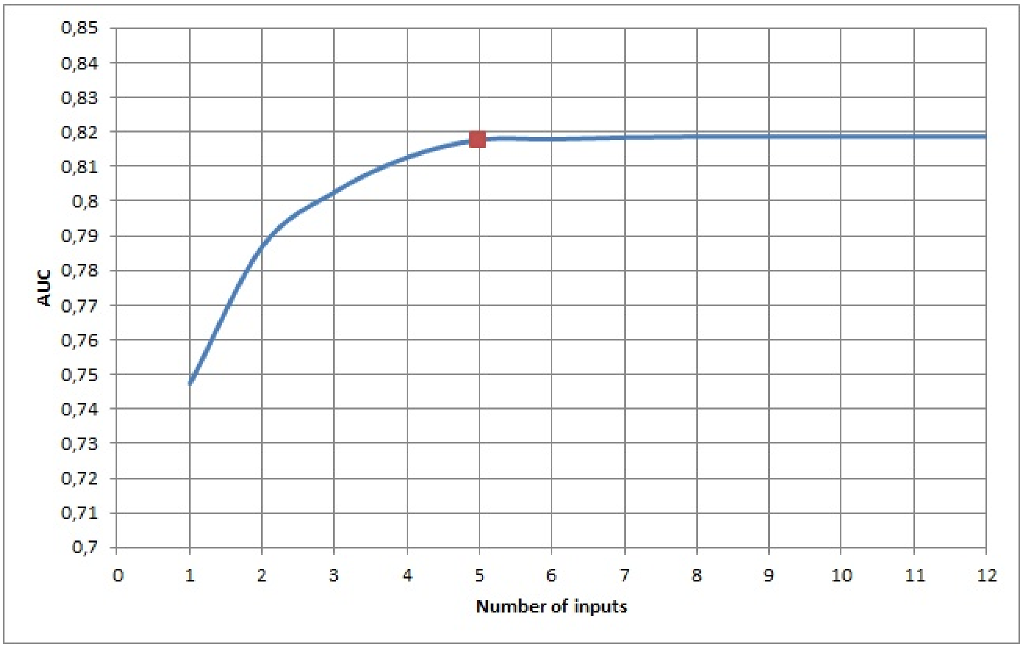

Figure 7. AUC for each model iteration applied to the HEART dataset.

some containing zero values for as much as 96% of the observations). For this reason, the dataset was altered by changing each input into a binary variable. Every binary variable was given a value of 0 if the entry of the original input was 0 and a value of 1 if its entry was greater than 0. The binary variables were named by adding the prefix *bin* in front of the original variable's name, e.g. if the original input was called $DOLLAR$, the binary version would be called bin_DOLLAR. Both models in SAS and CPLEX were then fitted using the binary predictors as inputs instead of the original variables. The only columns that remained unchanged (no binary indicators were created for them) were $CAPAG$, $CAPLONG$ and $CAPTOTAL$. These variables related to the running length of capital letters in the email and did not contain any zeros. Ultimately, each of the binary variables can therefore be seen as an indication of whether or not a certain word occurred within the mail.

Table 4. Coefficient estimates obtained for the HEART dataset during best subset selection.

Variable	CPLEX fitted coefficient	SAS fitted coefficient
Intercept	−8.8826	−8.8905
AgeAtStart	0.1096	0.1096
Systolic	0.0185	0.01849
Smoking	0.0301	0.03016
CHD_ind	1.1009	1.1023
Sex_bin	0.6170	0.6175

Table 5. Evaluation metrics for the HEART dataset based on models with best subset selection.

Evaluation metric	**CPLEX**	**SAS**
Train Gini	63.79	63.74
Test Gini	61.36	61.33
Train Accuracy	75.93%	75.85%
Test Accuracy	75.77%	75.61%

To determine an appropriate value for the cardinality parameter q, the same approach that was followed in Section 5.3 when best subset selection was carried out for the HEART dataset, was applied here. Once more, the best l-variable model, for $l = 1, \dots, p$, was iteratively produced in SAS using PROC LOGISTIC with the SELECTION = SCORE option, after which the AUC for each model fit was recorded. A suitable candidate for the cardinality constraint in best subset selection was determined by using the value of q that corresponds to the point at which no real improvement in AUC can be seen. Figure 8 shows that a model with roughly eight input variables should be adequate (refer to the square marker).

The linearised model with $q = 8$ plus one more to account for the intercept, employed a grid spanning $[-30, 30]$ with 0.01 intervals and its optimality gap was set to 5% in order to aid execution times. Both the linearised model and model in SAS produce solution vectors that are alike for the majority of the variables, noting that seven out of the eight non-zero fitted coefficients correspond to the same inputs. It can also be seen that the estimated regression coefficients obtained for these seven inputs are very close to one another within the two solution vectors, with minor differences existing between the two sets. The only exception is where SAS fits a non-zero regression estimate to the variable bin_YOUR and the linearised model in CPLEX chooses a non-zero coefficient for bin_OUR. While these are, in fact, different variables, both obtained estimated regression parameters approximately equal to one (with $\hat{\beta}_{bin\_OUR}$ being exactly equal to one). Table 6 displays the coefficients obtained by the two models.

When considering the accuracy of the models produced by the two respective techniques, it was found that the linearised model obtained a higher objective function value of −740.66 for the log-likelihood as opposed to the slightly lower −742.449 produced by its counterpart in SAS. The final model fitted by SAS had a fractionally higher Gini statistic of 92.13 for the training set, as opposed to a Gini of 92.09 associated with the linearised model with best subset selection. Alternatively, the CPLEX logistic regression model obtained marginally better results for the test dataset, with a Gini of 91.37 being higher than the 90.91 seen when SAS's regression model is applied to the same test set. In terms of misclassification rates, both models obtained an accuracy statistic of 91.06% for the training dataset. However, once again, the linearised model appeared to fair better than the model produced by SAS when test set accuracy was considered. While the model fitted in CPLEX showed a misclassification rate of 8.59% for the test dataset, SAS's final model obtained a misclassification rate of 9.38% for the same set of data. The very small discrepancies that exist between the Gini statistics and accuracy measures of the training and validation sets prove that both models appear to

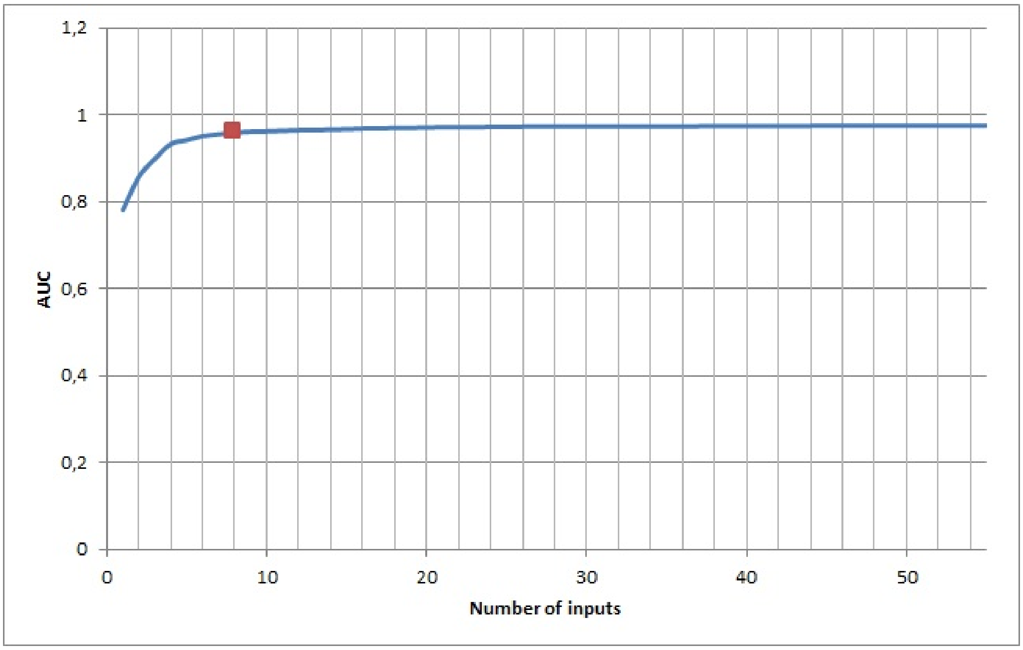

Figure 8. AUC for each model iteration applied to the JUNKMAIL dataset with binary predictors.

be quite stable.

In an attempt to understand why the two separate models are not producing identical solution vectors, due to their inclination to have at least one predictor that differs in the subset of variables included in the final model, both models were refitted again. During each instance of model fitting, a different value was specified for the cardinality parameter q. The initial value of q was set to be relatively low and subsequently increased in an ordinal fashion after each run.

It was noted that up until $q = 6$, both models yield identical solutions. At $q = 6$, the two models choose to include bin_DOLLAR , bin_EDU, $bin\_EXCLAMATION$, bin_FREE, bin_HP and bin_REMOVE. While small differences exist between the fitted regression parameter vectors produced by the respective models, the estimated coefficients for these six inputs were largely the same when comparisons were made. It should come as no surprise that both models achieved an objective function value for the log-likelihood as well as Gini values and misclassification rates for the training and test datasets that were nearly identical. At $q = 7$, SAS's logistic procedure opts to fit a non-zero coefficient to the variable bin_YOUR, whereas the linearised model in CPLEX chooses to include the variable $bin\_MEETING$ instead. When $q = 8$, the SAS logistic procedure concedes to choose $bin\_MEETING$ for inclusion in the final model (which was selected by the CPLEX model at $q = 7$ in the previous step). However, the CPLEX model still chooses not to fit a non-zero coefficient to the variable bin_YOUR (which was chosen by SAS at $q = 7$) and instead opts for the inclusion of the input bin_OUR (as was seen in Table 6). When the cardinality parameter was set to $q = 9$, it was found that both models elected the variable bin_1999 for inclusion at this step. Again, as with $q = 8$, the solution vector produced by SAS contained bin_YOUR in the final model, whereas

Table 6. Coefficient estimates obtained for the JUNKMAIL dataset.

SAS		CPLEX	
Variable	**Coefficient**	**Variable**	**Coefficient**
Intercept	−2.3365	Intercept	−2.2
bin_DOLLAR	2.173	bin_DOLLAR	2.3
bin_EDU	−3.1087	bin_EDU	−3.1
$bin\_EXCLAMATION$	1.54	$bin\_EXCLAMATION$	1.6
bin_FREE	1.5861	bin_FREE	1.6
bin_HP	−3.7084	bin_HP	−4
$bin\_MEETING$	−2.976	$bin\_MEETING$	−3.2
bin_REMOVE	2.757	bin_REMOVE	2.7
bin_YOUR	0.9179	bin_OUR	1

the CPLEX solution vector contained bin_OUR instead. Finally, at $q = 10$, the logistic regression model fitted in SAS selected the variable bin_OUR as an addition to the final model – an input that was selected by the CPLEX model much earlier on. However, the model produced by CPLEX still chooses to neglect the variable bin_YOUR, which was already selected by SAS's logisitic procedure at $q = 7$, and instead chooses the predictor bin_RE to be part of the final solution. Table 7 shows the log-likelihood values obtained along with model accuracy measures calculated for the training and test sets, respectively, for different values of the cardinality parameter q.

Table 8 displays the regression parameter estimates produced by the two models for different cardinality constraints.

Table 7 reveals that the logistic regression models fitted by the linearised model in CPLEX consistently produced log-likelihood values along with Gini and accuracy statistics that were better than those obtained by its SAS counterpart in most cases, regardless of the value of q.

Lastly, empirical results in Sato et al. (2015) show that the linearised approximation with variable selection proposed by the authors selected between 28 and 31 variables for inclusion in the final model when the approach in (10)–(13) was applied to the JUNKMAIL dataset. Unfortunately, the authors do not report on model evaluation metrics such as the misclassification rate, Gini statistic or AUC. Additionally, instead of splitting the modelling dataset so as to perform training versus validation exercises, Sato et al. (2015) chose to utilise the full JUNKMAIL dataset during model fitting. As a result, a comprehensive comparison of the different linearised approaches discussed in Sections 2 and 3 could not be carried out. However, it can be seen that the model in (10)–(13) selected more than three times the number of variables when compared the linearised model proposed in this paper. Furthermore, it is also observed that the eight-variable model shown above displays impeccable levels of accuracy while simultaneously maintaining a high level of descriptiveness by parsimoniously including a low number of predictors. In spite of the favourable results achieved in this section, it would be naive to assume that the differences that exist between the model fitted by Sato et al. (2015) and the model fitted above can entirely be attributed to the use of (18)–(24) instead of (10)–(13). Instead, it should be acknowledged that an indispensable amount of value is added to the quality of the models by exploratory data analysis and appropriate variable transformations.

Table 7. Model accuracy statistics for training and test sets for different values of q.

	SAS					CPLEX				
q	$\log L$	**Train Gini**	**Test Gini**	**Train Accuracy**	**Test Accuracy**	$\log L$	**Train Gini**	**Test Gini**	**Train Accuracy**	**Test Accuracy**
6	−796.74	90.59	90.08	90.11%	89.97%	−796.85	90.60	90.06	90.11%	89.97%
7	−774.50	91.35	91.67	90.34%	90.17%	−763.08	91.47	91.05	90.34%	90.17%
8	−742.45	92.13	90.91	91.06%	90.63%	−740.66	92.09	91.37	91.06%	91.41%
9	−723.87	92.62	91.65	91.58%	90.95%	−720.95	92.49	92.02	91.42%	91.80%
10	−708.72	92.67	91.67	91.32%	91.67%	−705.35	92.93	92.89	91.81%	91.60%

Table 8. Estimated regression coefficients obtained for different values of q.

	SAS					CPLEX				
Variable	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$
Intercept	−2.1601	−2.4335	−2.3365	−2.2671	−2.335	−2.2	−2.1	−2.2	−2.15	−2.0333
bin_DOLLAR	2.3756	2.1352	2.173	2.2508	2.1846	2.4	2.4	2.3	2.35	2.5
bin_EDU	−3.3352	−3.4033	−3.1087	−2.989	−2.9616	−3.3	−3	−3.1	−2.9	−2.7
bin_EXCLAMATION	1.689	1.5476	1.54	1.5399	1.4633	1.7	1.7	1.6	1.55	1.6333
bin_FREE	1.6817	1.4807	1.5861	1.6956	1.5818	1.7	1.8	1.6	1.75	1.8
bin_HP	−3.7405	−3.7459	−3.7084	−3.1803	−3.3536	−3.7	−3.7	−4	−3.4	−3.4667
bin_REMOVE	3.0102	2.8466	2.757	2.7048	2.5401	3	2.9	2.7	2.65	2.6333
bin_YOUR		0.9324	0.9179	0.9173	0.7353					
bin_MEETING			−2.976	−2.8226	−2.9963		−3	−3.2	−3.05	−3.0667
bin_OUR					0.8855			1	1.05	1.1
*bin_*1999				−1.4727	−1.5279				−1.55	−1.5333
bin_RE										−0.9667

5.5 Test on high dimensional data

For this exercise, best subset selection was performed on a dataset where the number of input variables is substantially large. By using the same approach described in Section 5.1, a dataset containing $n = 1\ 000$ cases and $p = 500$ standard normal random variables was generated. Of the 500 effects, a total of 15 equally spaced predictors were chosen to influence the response and have true regression coefficients equal to one, meaning that $\beta_l^* = 1$ where $l \in Q$ and $Q = \{1, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121, 131, 141\}$. The remaining 485 inputs were specified to have $\beta_j^* = 0$ with $j \notin Q$ and had no affect on the response.

As stated previously, best subset selection becomes extremely resource intensive when $p \gg 40$ due to the exponential increase in execution time for each additional variable considered by the model. For this reason, both models in SAS and CPLEX were only allowed to execute model fitting for a predetermined amount of time. Once again, the model in (18)–(24) was used to facilitate best subset selection in CPLEX, while the SELECTION = SCORE with START = 15 and STOP = 15 options were specified in SAS PROC LOGISTIC.

After only two hours of execution, the linearised model in CPLEX already selects the correct 15 variables for inclusion by setting $z_l = 1$ where $l \in Q$. Furthermore, all other variables are omitted from the final model due to $z_j = 0$, $j \notin Q$, in the CPLEX solution. Terminating the CPLEX logistic regression model after two hours yields a log-likelihood value of −276.762 with an optimality gap of 78.70%[7]. The exact same model is obtained when CPLEX is allowed to terminate after four hours instead of two, with an identical objective function value of −276.762. However, the optimality gap is reduced to 71.35%. In an attempt to find a model that is closer to optimality, the linearised model in CPLEX was applied once more to the data with a specified run time of 48 hours. As with the models produced after two and four hours, respectively, coefficient estimates greater than zero are obtained for the correct 15 variables, while all other parameter estimates are set equal to exactly zero. Once again, a maximum likelihood estimate of −276.762 is produced. In spite of identical log-likelihood values, a significantly lower optimality gap of 58.68% is observed. Table 9 shows the parameter estimates obtained by the linearised logistic regression model in CPLEX. Investigation revealed that all three of the aforementioned models yielded the same estimated coefficient values regardless of the execution times that were specified. Notice that the regression coefficients which were estimated for the 15 variables are sufficiently close to their true underlying values of one.

In contrast to the linearised model in CPLEX, SAS fails to yield any results after 48 hours of execution time. However, this can be expected, as best subset selection in SAS is considered to be computationally impractical when $p > 75$ (Lund, 2017). Several course notes on regression modelling in SAS, such as Patetta (2012) and Patetta, Huber and Nizam (2009), suggest that modellers should opt for stepwise variable selection procedures in such cases instead.

The results indicate that the linearised model yields sufficient output and can adequately perform best subset selection when the number of predictors p is large. Specifically, the model in (18)–(24) tends to select the appropriate inputs and produces the correct underlying model even when execution is stopped early. Empirical studies show that (18)–(24) finds the correct model and parameter estimates early on, but requires significantly more time to prove optimality. This is supported by the

[7] The optimality gap refers to the relative disparity that exists between the best lower bound solution and the best upper bound solution, expressed as a percentage.

Table 9. Parameter estimates produced by the linearised model in CPLEX for $p = 500$ and $q = 15$.

Coefficient	**Estimate**
β_0	0.0137
β_1	1.2034
β_{11}	1.1285
β_{21}	0.9799
β_{31}	1.1253
β_{41}	0.9263
β_{51}	0.8725
β_{61}	1.1161
β_{71}	0.9909
β_{81}	1.1440
β_{91}	1.0676
β_{101}	1.1994
β_{111}	1.1556
β_{121}	1.1341
β_{131}	0.9378
β_{141}	0.9505

fact that all CPLEX runs obtained identical log-likelihood objective function values and estimated coefficients, but exhibited a declining optimality gap alongside an increase in run time. Bertsimas et al. (2016) encountered the same phenomenon, noting that their suggested MIO formulation for best subset selection in linear regression produced suitable models fairly quickly, but needed notably more time to provide a guarantee on optimality.

5.6 Choice of grid range and number of grid values, k

When the linearised model was fitted to both simulated and real-world datasets, it became apparent that the range specified for the grid used in the linearisation is of a subjective nature and problem-specific. Indeed, much wider ranges needed to be used for the JUNKMAIL dataset where the estimated coefficients could potentially take on values much larger than one, as opposed to simulated data where the inputs were standardised. However, it was found that if the true underlying regression parameters are close to one or relatively small, the following range for the grid can be employed:

$$[-(1+b)q, (1+b)q], \tag{25}$$

where q is the number of variables that have an influence on the response with the true underlying regression coefficients being non-zero. If q is unknown, it can be replaced with p. The parameter b is a buffer used to allow the model to deviate somewhat from the values of the true underlying coefficient parameters when fitting the model, i.e. to allow for white noise. During the exercises conducted on simulated data (as discussed in Sections 5.1 and 5.2), $b \leq 1$ was found to be more than adequate.

Alternatively, if no indication exists as to what values the true underlying regression parameters may take on (as is the case with most real-world applications), a possible starting point would be to fit the full logistic regression model without variable selection (the choice of model does not matter, so either the linearised model in (14)–(17) or nonlinear model in (5) with Newton–Raphson method can be employed). Once the parameter estimates are obtained, calculate $M_1 = \min\{\sum_{l=1}^{p+1} \hat{\beta}_l X_{1l}, \ldots, \sum_{l=1}^{p+1} \hat{\beta}_l X_{nl}\}$ and $M_2 = \max\{\sum_{l=1}^{p+1} \hat{\beta}_l X_{1l}, \ldots, \sum_{l=1}^{p+1} \hat{\beta}_l X_{nl}\}$, where n is the number of observations in the modelling set. The following incarnation of (25) can then be used as a potential range for the grid:

$$[M_1 - b_1, M_2 + b_2], \tag{26}$$

where b_1 and b_2 have the same interpretation as b in (25) and are not necessarily equal to one another (however, they might be). Note that when (26) is used, slightly larger values for b_1 and b_2, with $b_1 > 1$ and $b_2 > 1$, need to be specified. For the real-world datasets presented in this paper, $5 \leq b_{opt} \leq 20$ with $b_{opt} = b_1 = b_2$ was found to be a suitable choice. Occasionally, a situation may present itself where M_1 or M_2 (or both) is unreasonably large. This is often caused by a select few outliers that are present in the data. If M_1 or M_2 is significantly inflated, it may lead to extended model execution times (if more grid values are used) or a linearised logistic regression model that severely underestimates the nonlinear log-likelihood (if k is not increased to compensate for the wider grid). To combat this, the user may opt to utilise $M_1^{\alpha} = P_{\alpha}$ and $M_2^{\alpha} = P_{(1-\alpha)}$ instead, where P_{α} denotes the $(\alpha \times 100)$-th percentile of the set $\{\sum_{l=1}^{p+1} \hat{\beta}_l X_{1l}, \ldots, \sum_{l=1}^{p+1} \hat{\beta}_l X_{nl}\}$. While $\alpha = 0.05$ serves as a common middle ground, more conservative modellers might consider using $\alpha = 0.01$, whereas $\alpha = 0.1$ may be employed by users who are less concerned with model accuracy and more focussed on speed of execution.

The number of grid values k used in (18)–(24) remains a more subjective issue. As mentioned previously, as k tends to infinity, the linearised model resembles the nonlinear log-likelihood function. Therefore, increasing k should result in improved estimates and, subsequently, a more accurate model. However, experimental results have revealed that a substantial increase in k results in a rapid increase in model execution time, but does not necessarily deliver a significant improvement in model fit.

The results presented next in Section 6 examine the trade-off that exists between the quality of the solutions produced by the MILP formulation and model execution time for different values of k.

6. Improving computing times

As stated before, best subset selection is an NP-hard problem which is often considered to be resource intensive and is regularly abandoned in favour of more computationally friendly approaches, such as stepwise regression. However, the results in this section will demonstrate that the application of the MILP formulation in (18)–(24) with the appropriate settings allows the modeller to obtain solutions that are closer to optimality within an increasingly shorter time frame, even in the case of large datasets. In fact, improvements in model run time and/or optimality can be achieved by simply specifying a suitable value for the number of grid values k utilised by the linearised approximation of the log-likelihood function and by considering the optimality gap of the MILP prior to execution.

By specifying that the optimality gap must be greater than zero, significantly improved execution times can be achieved. In Section 5 the optimality gap was briefly addressed. The gap refers to the difference between the best lower bound solution and the best upper bound solution, which is subsequently expressed as a percentage. The default optimality gap in the mathematical solver CPLEX is usually set equal to a very small number that is close to zero, such as $1e - 5$. A gap of zero will mean that the solution yielded by the model can be considered as the most optimal possible solution attainable and cannot be improved. However, a smaller optimality gap that is closer to zero might result in a situation where the solver takes longer to terminate, which in turn can lead to prolonged model execution times. Ultimately, extended model run times in the presence of a small gap can be attributed to the fact that the solver has to evaluate more nodes during its execution in order to prove optimality. Empirical evidence in Section 5 of this paper, as well as results presented by Bertsimas et al. (2016), suggest that the MILP regression model finds the correct solution fairly quickly during the early stages of model fitting (when the gap is still relatively large), but requires additional time in order to prove optimality. Therefore, the user can comfortably request the model to terminate at a larger optimality gap, knowing that the resulting solution will be close to optimality (or may even be the optimal solution). All of the results presented in this section were obtained from models that were executed where the gap was set equal to 5% (unless stated otherwise), meaning that the solutions found by these models are at most 5% worse than the most optimal solution. For a detailed explanation on the concept of the optimality gap in mixed integer problems, refer to Gurobi Optimization (2017).

Finally, recent advances in computing power, which have enabled solvers to take on much larger and more complex problems, should also be acknowledged.

In this section, experiments were performed by fitting the linearised model in (18)–(24) to the data for different values of k in order to investigate the trade-off between model run time and quality of the results produced. Specifically, logistic regression models were iteratively executed on the same set of data for $k \in \{401, 161, 81, 41\}$. It should be noted that the maximum number of grid values employed by all models in this section was set equal to 401 and that the solutions obtained when $k = 401$ were used as a baseline reference. This means that the execution times of and results yielded by all other models that were fitted to the same dataset for $k \neq 401$ were compared to those found for the baseline model. This is motivated by the results obtained in Section 5, where it was found that the linearised logistic regression model with $k = 401$ produced maximum likelihood estimates that were sufficiently close to the MLE obtained by the nonlinear model in SAS. Additionally, it was observed that using much larger values for k unnecessarily increased model execution times without showing significant improvement in model fit.

Consider once more the simulated datasets that were used in the modelling exercises conducted in Section 5.1 and recall that various training sets containing n observations, with $n \in \{1\,000, 3\,000, 5\,000, 10\,000\}$, and 100 variables from a standard normal distribution were generated. In all of these datasets, 10 equally spaced features were chosen to influence a latent, continuous response variable $\mathbf{Y}^c$ and were specified to have a true regression coefficient of one, or $\beta_l^* = 1$ for $l \in Q$, where $Q = \{1, 11, 21, 31, 41, 51, 61, 71, 81, 91\}$. A binary outcome vector was subsequently created from $\mathbf{Y}^c$ and served as the target variable for the logistic regression model. By knowing that the final model should contain 10 predictors, the model in (18)–(24) was fitted to the data whereby q was set equal to 10 plus one more to compensate for the intercept. Table 10 displays the maximum likelihood estimates obtained during each round of model fitting for different values of k, along with the corresponding time consumed by every instance of model execution.

Table 10 shows that a significant decrease in model execution time can be achieved by reducing k. However, shrinking the number of grid values employed by the linearised model does not appear to have an adverse effect on the MLE obtained by the model. Table 11 shows by how much the

Table 10. Linearised logistic regression model performance for different values of k.

	$\log L$				**Time (in seconds)**			
k	$n = 1\,000$	$n = 3\,000$	$n = 5\,000$	$n = 10\,000$	$n = 1\,000$	$n = 3\,000$	$n = 5\,000$	$n = 10\,000$
401	−321.037	−1 010.981	−1 755.168	−3 500.658	1 727.93	3 231.42	25 148.31	102 583.00
161	−321.459	−1 012.392	−1 757.545	−3 505.505	394.76	1 803.31	5 060.53	9 531.11
81	−323.021	−1 017.388	−1 766.271	−3 522.718	217.31	810.53	1 964.24	6 657.11
41	−329.252	−1 037.018	−1 800.191	−3 589.838	99.75	322.35	1 105.39	5 467.77

Table 11. Deterioration in $\log L$ for different values of k.

k	$n = 1\,000$	$n = 3\,000$	$n = 5\,000$	$n = 10\,000$
401	–	–	–	–
161	−0.13%	−0.13%	−0.13%	−0.13%
81	−0.61%	−0.63%	−0.63%	−0.63%
41	−2.55%	−2.57%	−2.56%	−2.54%

optimum solution has weakened (in percentage terms) alongside smaller k values. It can be observed that specifying lower values for k seems to exert very little influence on the quality of the solution obtained.

From the results displayed in Tables 10 and 11 it can be seen that the amount of time required to perform logistic regression modelling with best subset selection may be reduced by up to 90% if the modeller is willing to accept a final solution that is less than 3% weaker. Furthermore, it should be noted that all of the models selected the correct 10 variables for inclusion in the final solution, regardless of the value specified for k. Lastly, parameter estimates obtained for these 10 inputs were sufficiently close to their true underlying values of one, i.e. $\hat{\beta}_l \approx 1$ for $l \in Q$, after each round of model fitting.

Modelling exercises similar to those presented above were conducted on real-world datasets as well. Note that model performance for the HEART dataset in Section 5.3 was not considered in this section, as the modelling set contains only 12 potential predictor variables, which can easily be accommodated by most variable selection techniques. Instead, two additional real-world datasets were studied alongside the JUNKMAIL dataset, namely the ONLINE NEWS POPULARITY dataset of Fernandes, Vinagre and Cortez (2015) and the SUPERCONDUCTIVITY DATA dataset of Hamidieh (2018). Both datasets, along with the JUNKMAIL dataset, were chosen for inclusion in this section of the paper due to their relative size. The number of rows, n, as well as the dimensionality, p, of these datasets are sufficiently large and are more representative of an actual set of data that a modeller might encounter in practice. All of the aforementioned data are available for download from the UCI Machine Learning Repository.

The ONLINE NEWS POPULARITY dataset contains a total of $n = 39\ 640$ records and 61 columns, of which 58 are potential predictors. Of these 58 columns, four were found to be linear combinations of other features in the dataset and were subsequently removed, leaving 54 variables that were eligible for inclusion in the final model. The target variable in question is called *Shares* and captures the number of online shares of a news article upon publication. It was noted that the response vector is a numeric field which contains integer entries, where $Shares \in \mathbb{Z}^+$, which implies that it cannot be used as-is in binary classification problems such as logistic regression. Instead, a suggestion given by Fernandes et al. (2015) was applied whereby a news article was classified as popular when it had more than 1 400 shares. Alternatively, an article with a lower number of shares was seen as unpopular. This lead to the creation of the binary outcome variable **Y** for logistic regression by specifying

$$Y_i = \begin{cases} 0 & \text{if } Shares < 1400, \\ 1 & \text{if } Shares \geq 1400, \end{cases}$$

for $i = 1, \ldots, n$.

The final dataset contained a total of 18 490 records with $Y = 0$ and 21 150 cases with $Y = 1$ after the target variable was transformed. As is the case within most predictive modelling settings, the entire dataset containing almost 40 000 observations was not used for model training. A smaller training set consisting of $n_1 = 10\,000$ cases was created by randomly sampling 5 000 observations with $Y = 1$ and 5 000 observations with $Y = 0$ from the total set. The remaining $n_2 = 29\,640$ were used as a validation set to test the generalisation abilities of the model. Lastly, studying the AUC curve produced by models fitted to the dataset revealed that the model becomes saturated when approximately 22 predictors are included in the final solution. Therefore, the model in (18)–(24) with best subset selection was executed on the training set by using $q = 22 + 1$ as a cardinality parameter (one more to compensate for the intercept term in regression)[8].

The SUPERCONDUCTIVITY DATA dataset contains information relating to the chemical compound of superconductors, along with the critical temperature achieved by the superconductor which is captured in a column named *critical_temp* and serves as the response variable of interest. The table comprises of $n = 21\,263$ observations and 81 features that can be used as potential predictors in the final model. As is the case with the ONLINE NEWS POPULARITY dataset, the outcome variable had to be discretised in order to formulate a suitable target variable for logistic regression, since the field *critical_temp* is a postie, real-valued number. In a separate study wherein machine learning models were trained to predict the critical temperature of superconductors, Stanev, Oses, Kusne, Rodriguez, Paglione, Curtarolo and Takeuchi (2018) note that superconductors can be adequately divided into two groups of materials associated with either high or low temperatures by using *critical_temp* = 10K as a cut-off. A binary response variable was subsequently created as follows:

$$Y_i = \begin{cases} 0 & \text{if } critical\_temp < 10, \\ 1 & \text{if } critical\_temp \geq 10, \end{cases}$$

for $i = 1, \ldots, n$.

Once the transformation shown above was carried out, the final dataset encompassed a total of 7 729 observations with $Y = 0$ and 13 534 records with $Y = 1$. Once more, a smaller dataset was created for the purposes of training the model. Results presented by Stanev et al. (2018) show that satisfactory levels of accuracy can be achieved when predicting superconductor temperatures with a modelling set that contains roughly 3 000 to 6 000 observations. For this reason, a training dataset

[8] Recall that the same methodology was used to select a suitable value for q when logistic regression modelling with best subset selection was carried out for the HEART and JUNKMAIL datasets in Section 5.

with $n_1 = 6\,000$ observations was composed by randomly sampling 3 000 cases with $Y = 0$ and 3 000 records with $Y = 1$. The other $n_2 = 15\,263$ observations that were excluded from the training set were kept as a test set. In order to decide on a suitable value for the cardinality parameter q, consideration was given to the results presented by Hamidieh (2018) and Stanev et al. (2018) instead of having to draw up and analyse the AUC curve for 81 variables. Hamidieh (2018) opts to fit an extreme gradient boost model to the data, where the untransformed response variable *critical_temp* is used as a target. The author then proceeds to report the objective function gains achieved for each input variable as a percentage of the total objective function improvement obtained when all features are included in the model. By studying the individual gains of the top 20 predictors in the model, it was found that the improvement in the final objective function value seems to diminish between the first 10 to 15 variables. In turn, the emprical evidence supplied by Stanev et al. (2018) indicate that the performance of their classifiers achieve a plateau for a model consisting of approximately 10 or more predictors. After analysing the results presented by the aforementioned authors, a value of $q = 10 + 1$ was used when performing best subset selection on the superconductors dataset in order to obtain a final model with 10 inputs and an intercept.

Recall that the exercises conducted in Section 5.4 involved the execution of the linearised logistic regression model with best subset selection for the JUNKMAIL dataset where a variety of values were specified for the cardinality parameter q, which produced solutions containing between $q = 6$ and $q = 10$ variables in the final model. This exercise was repeated, with the difference being that increasingly less knot values were used for the linearisation of the log-likelihood objective function. Table 12 displays the objective function value obtained and time consumed during each round of model fitting, whereas Table 13 shows by how much the final solution has weakened alongside any potential improvements in model execution time.

From Tables 12 and 13, it can be seen that the use of a rougher grid (i.e. less grid values or a smaller k) results in a tremendous improvement in model execution time. Alternatively, no real deterioration in the final log-likelihood objective function values is evident. Finally, while not displayed here, note that Gini measures and accuracy rates similar to those in Table 7 were found for all of the aforementioned models that were fitted to the JUNKMAIL dataset, regardless of the size of the grid.

As is the case with the JUNKMAIL dataset, empirical evidence suggests that the user sacrifices very little when it comes to the quality of the final model when a less granular grid is utilised during fitting of logistic regression models to the ONLINE NEWS POPULARITY dataset. However, significant gains can be made with respect to model run times when the number of grid values k is smaller. This is supported by results presented in Table 14 where the MILP formulation in (18)–(24) was tasked with fitting a 22-variable model for a variety of values specified for k.

By studying Table 14, notice that all of the models that were fitted to the ONLINE NEWS POPULARITY dataset obtained Gini values between 37 and 41. In turn, a Gini of 37 to 41 translates to an AUC measure of approximately 68 to 71. These measurements closely resemble the model evaluation metrics associated with a random forest model and gradient boosting machine that was fitted to the same set of data by Fernandes et al. (2015). The random forest obtained by the authors had an AUC of 72.7, while the gradient boosting machine achieved an AUC of 74.5. This suggests that a rudimentary classifier like the logistic regression MILP, which produces simpler models that are arguably more interpretable, has the ability to yield final models with similar predictive performance when compared to much more complicated machine learning algorithms, such as random forests and

Table 12. Model results for different values of k for the JUNKMAIL dataset.

	$\log L$					**Time (in seconds)**				
k	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$
401	−796.853	−763.082	−740.659	−720.951	−705.347	3 012.08	5 479.64	8 104.39	11 897.68	22 255.88
161	−797.393	−764.197	−741.289	−722.038	−705.893	866.62	1 615.34	2 293.23	3 696.87	6 229.48
81	−799.669	−765.184	−744.089	−723.094	−707.086	366.51	610.10	859.92	1 516.28	2 308.04

Table 13. Deterioration in log-likelihood and improvements in run times for the JUNKMAIL dataset

	Deterioration in $\log L$					**Reduction in run time**				
k	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$	$q = 6$	$q = 7$	$q = 8$	$q = 9$	$q = 10$
401	–	–	–	–	–	–	–	–	–	–
161	−0.06%	−0.14%	−0.08%	−0.15%	−0.07%	−71.22%	−70.52%	−71.70%	−68.92%	−72.01%
81	−0.35%	−0.26%	−0.46%	−0.30%	−0.25%	−87.83%	−88.87%	−89.39%	−87.26%	−89.63%

Table 14. Model results for different values of k for the ONLINE NEWS POPULARITY dataset.

k	$\log L$	**Time (in seconds)**	**Train Gini**	**Test Gini**	**Train Accuracy**	**Test Accuracy**
401	−6 298.610	9 610.88	40.92	37.60	64.87%	63.92%
161	−6 320.899	3 139.32	40.29	37.51	64.93%	63.80%
81	−6 333.254	1 364.88	40.71	37.29	65.09%	63.52%

Table 15. Model results for different values of k for the SUPERCONDUCTIVITY DATA dataset.

k	$\log L$	**Optimality Gap**	**Train Gini**	**Test Gini**	**Train Accuracy**	**Test Accuracy**
401	−2 038.729	22.57%	85.46	85.68	85.10%	84.49%
161	−1 977.003	20.02%	86.49	86.67	85.20%	84.96%
81	−1 955.179	20.37%	86.27	86.51	86.07%	86.04%
41	−1 969.546	17.26%	87.33	87.49	86.57%	86.67%

gradient boosting machines.

The SUPERCONDUCTIVITY DATA dataset is by far the largest real-world dataset considered within the best subset variable selection problem setting presented in this paper. Not only does the training set consist of many rows (n_1 = 6 000 cases), but it contains 81 potential predictors of various scales, which is about twice the number of variables that best subset selection procedures are expected to handle efficiently. For this reason, all logistic regression models that were fitted to the SUPERCONDUCTIVITY DATA were allowed to execute for a maximum of 86 400 seconds, which translates to 24 hours. Table 15 summarises the results obtained during each round of model fitting when the logistic regression MILP was used to obtain 10-variable solutions.

Even though none of the models were able to conclude their execution within the specified time limit, Table 15 suggests that high quality solutions are still achievable even when model fitting is stopped early. Specifically, notice that improvements in the final log-likelihood objective function value and optimality gap are realised when the MILP formulation employs a grid that is more coarse. This is supported by consistently higher Gini and accuracy measures associated with lower k values. This can most likely be attributed to the fact that the solver has to consider fewer constraints within the limited amount of time that is made available for model fitting.

The results presented in this section indicate that the use of a less granular grid when performing best subset selection with the model in (18)–(24) entails many benefits. For moderately large datasets, such as the JUNKMAIL dataset and ONLINE NEWS POPULARITY dataset, considerably improved model run times can be achieved if the modeller is willing to accept an insignificant decay in the quality of the final solution. For much larger datasets, like the SUPERCONDUCTIVITY DATA dataset, the use of a grid consisting of less knot values can still yield satisfactory models alongside potential improvements in the optimality gap and overall model fit, even when model execution terminates early.

7. Conclusion and future work

Rapid advances in computing power and algorithmic execution in recent times have lead to a resurgence in interest in research directed towards the optimisation of linear models using exact approaches. In this paper, the estimation of the parameter vector in a logistic regression model using a linearised approximation of the log-likelihood function was shown. The proposed linearised logistic regression model yields a solution that underestimates the maximum likelihood estimate. Experimental results indicate that the linearised model produces results that are comparable (or identical) to those obtained by models fitted using the iterative Newton–Raphson method. In some cases, the linearised model performed better than its iterative counterpart. The suggested model (without subset selection) yields solutions within a reasonable amount of time when the number of predictors, p, is in the 100s and the number of observations, n, is in the 1000s, noting that scalable solutions are still attainable even when n is notably larger at 20 000 records.

The proposed model was also extended to facilitate best subset selection via the introduction of cardinality parameters. The results presented in this paper show that the linearised logistic regression model with best subset selection consistantly produces parsimonious final models that are similar to or occasionally better than the benchmark. Furthermore, it is observed that a trade-off exists between accuracy and execution time when considering the number of grid values, k, employed in the linearisation of the log-likelihood objective function. Empirical evidence provided in Section 6 show that the logistic regression MILP with best subset selection can produce models within increasingly shorter periods of time if a suitable value for k is specified. In fact, by shrinking k quite significantly, the modeller is able to obtain solutions considerably quicker without having to sacrifice much with respect to the quality of the solution achieved. For substantially larger datasets, the use of less grid values still yields satisfactory logistic regression models, even when model execution is stopped early. Additionally, potential improvements in overall model fit and accuracy measures can be realised when k is reduced for sizable modelling sets. A suggested approach on how to select an appropriate grid for the linearised model was presented. While the approach put forth in Section 5 for selecting suitable start and end points for the grid is relatively user-friendly and requires little mathematical rigour, alternative methodologies towards the specification of the grid can perhaps be explored.

It is noted that the suggested model with best subset selection tends to find the best q-variable model and its corresponding parameter estimates fairly quickly, but requires a significant amount of time to prove optimality. In turn, this can result in increased execution times. However, this is not unique to the model suggested in (18)–(24) and has been documented for best subset selection applications in linear regression as well – see Bertsimas et al. (2016). Therefore, further consideration should be given to the structure of the MILP constraint matrix and the decomposition thereof for additional algorithmic advances and reduced model run times.

Lastly, the properties of the logistic regression estimate vector $\hat{\beta}$ should be explored to a greater extent. Specifically, the effect of the upper and lower bounds M_{lL} and M_{lU} in constraint (22) in (18)–(24) and the size of the grid should be investigated, as significantly lower upper bounds, higher lower bounds or a narrower grid has the potential to shrink fitted regression coefficients, which could yield biased estimates.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *In* Petrov, B. N. and Csaki, F. (Editors) *Proceedings of the 2nd International Symposium on Information Theory*. Budapest, 267–281.

Bertsimas, D. and King, A. (2016). OR Forum – an algorithmic approach to linear regression. *Operations research*, **64**, 2–16.

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

Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. *In Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence*. Portugal.

Furnival, G. and Wilson, R. (1974). Regression by leaps and bounds. *Technometrics*, **16**, 499–511.

Gurobi Optimization (2017). Gurobi Optimization User Manual: Mixed Integer Programming (MIP) – A Primer on the Basics. http://www.gurobi.com/resources/getting-started/mip-basics. Accessed on 2018-03-26.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, **154**, 346–354.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *Elements of Statistical Learning*. Springer Science and Business Media, New York.

Kamiya, S., Miyashiro, R., and Takano, Y. (2019). Feature subset selection for the multinomial logit model via mixed-integer optimization. *In Proceedings of Machine Learning Research, PMLR*, volume 89. 1254–1263.

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

Lund, B. (2017). Logistic model selection with SAS PROC's LOGISTIC, HPLOGISTIC, HP-GENSELECT – Paper AA02. *In Proceedings of the 2017 Midwest SAS Users Group Conference*. St. Louis.

Maldonado, S., Perez, J., Weber, R., and Labbe, M. (2014). Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, **279**, 163 – 175.

Miller, A. (2002). *Subset Selection in Regression*. CRC Press, Boca Raton.

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

Okeh, U. and Oyeka, I. (2013). Estimating the Fisher's scoring matrix formula from logistic model. *American Journal of Theoretical and Applied Statistics*, **2**, 221–227.

Patetta, M. (2012). *Predictive Modelling Using Logistic Regression Course Notes. Course code LWPMLR93/PMLR93*. SAS Institute Inc., Cary.

Patetta, M., Huber, M., and Nizam, A. (2009). *Categorical Data Analysis Using Logistic Regression Course Notes. Course code LWCDAL92/CDAL92*. SAS Institute Inc., Cary.

Potts, W. and Patetta, M. (1999). *Logistic Regression Modelling*. SAS Institute Inc., Cary.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Principles in FORTRAN: The Art of Scientific Computing*. Second edition. Cambridge University Press.

Rezac, M. and Rezac, F. (2011). How to measure the quality of credit scoring models. *Journal of Economics and Finance*, **61**, 486–507.

SAS Institute Inc. (2017). *SASHELP Data Sets*. SAS Institute Inc., Cary.

Sato, T., Takano, Y., Miyashiro, R., and Yoshise, A. (2015). Feature subset selection for logistic regression via mixed integer optimization. *Discussion Paper Series No. 1324, Department of Policy and Planning Sciences, University of Tsukuba*.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–464.

Stanev, V., Oses, C., Kusne, A. G., Rodriguez, E., Paglione, J., Curtarolo, S., and Takeuchi, I. (2018). Machine learning modeling of superconducting critical temperature. *Computational Materials*, **4**, 1–14.

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

Manuscript received 2019-05-02, revised 2019-09-20, accepted 2019-11-01.