Computational Optimization and Applications (2021) 80:1–32
https://doi.org/10.1007/s10589-021-00288-1

An effective procedure for feature subset selection in logistic regression based on information criteria

Enrico Civitelli[1] · Matteo Lapucci[1] · Fabio Schoen[1] · Alessio Sortino[1]

Received: 3 October 2020 / Accepted: 5 June 2021 / Published online: 17 June 2021

Abstract

In this paper, the problem of best subset selection in logistic regression is addressed. In particular, we take into account formulations of the problem resulting from the adoption of information criteria, such as AIC or BIC, as goodness-of-fit measures. There exist various methods to tackle this problem. Heuristic methods are computationally cheap, but are usually only able to find low quality solutions. Methods based on local optimization suffer from similar limitations as heuristic ones. On the other hand, methods based on mixed integer reformulations of the problem are much more effective, at the cost of higher computational requirements, that become unsustainable when the problem size grows. We thus propose a new approach, which combines mixed-integer programming and decomposition techniques in order to overcome the aforementioned scalability issues. We provide a theoretical characterization of the proposed algorithm properties. The results of a vast numerical experiment, performed on widely available datasets, show that the proposed method achieves the goal of outperforming state-of-the-art techniques.

Keywords Logistic regression · Information criterion · Best subset selection · Sparse optimization · Block coordinate descent

1 Introduction

In statistics and machine learning, binary classification is one of the most recurring and relevant tasks. This problem consists of identifying a model, selected from a hypothesis space, able to separate samples characterized by a well-defined set of numerical features and belonging to two different classes. The fitting process is based on a finite set of samples, the training set, but the aim is to get a model which correctly labels unseen data.

✉ Matteo Lapucci
matteo.lapucci@unifi.it

[1] Department of Information Engineering, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Florence, Italy

Among the various existing models to perform binary classification, such as k-nearest-neighbors, SVM, neural networks or decision trees (for a review of classification models see, e.g., the books of [9, 22] or [24]), we consider the logistic regression model. Logistic regression belongs to the class of Generalized Linear Models and possesses a number of useful properties: it is relatively simple; it is readily interpretable (since the weights are linearly associated to the features); outputs are particularly informative, as they have a probabilistic interpretation; statistical confidence measures can quickly be obtained; the model can be updated by simple gradient descent steps if new data are available; moreover, in practice it often has good predictive performance, especially when the size of train data is too limited to exploit more complex models.

In this work, we are interested in the problem of best features subset selection in logistic regression. This variant of standard logistic regression requires to find a model that, in addition to accurately fitting the data, exploits a limited number of features. In this way, the obtained model only employs the most relevant features, with benefits in terms of both performance and interpretation.

In order to compare the quality of models that exploit different features, i.e., models with different complexity, goodness-of-fit (GOF) measures have been proposed. These measures allow to evaluate the trade-off between accuracy of fit and complexity associated with a given model. Among the many GOF measures that have been proposed in the literature, those based on information criteria (IC) such as AIC [1], BIC [39] or HQIC [21] are some of the most popular [23]. Models based upon these Information Criteria are very popular in the statistics literature. Of course it is evident that no model is perfect and different models might had been considered. However, as nicely reported by [13], a reasonable model should be computable from data as well as based on a general statistical inference framework. This means that “model selection is justified and operates within either a likelihood or Bayesian framework or within both frameworks”. So, although many alternative models can be proposed and successfully employed, like, e.g., those described in [6, 8, 26], in this paper we prefer to remain on the classical ground of Information Criteria like the AIC, which is an asymptotically unbiased estimator of the expected Kullback–Leibler information loss, or the BIC, which is an easy to compute good approximation of the Bayes factor.

In case the selection of the model is based on one of the aforementioned IC, the underlying optimization problem consists of minimizing a function which is the sum of a convex part (the negative log-likelihood) and a penalty term, proportional to the number of employed variables; it is thus a sparse optimization problem.

Problems of this kind are often solved by heuristic procedures [17] o by ℓ_1-regularization [27, 29, 45]. In fact, specific optimization algorithms exist to directly handle the zero pseudo-norm [4, 31, 32]. However, none of the aforementioned methods is guaranteed to find the best possible subset of features under a given GOF measure.

With problems where the convex part of the objective is simple, such as least squares linear regression, approaches based on mixed-integer formulations allow to obtain certified optima, and have thus had an increased popularity in recent years [7, 14, 19, 34, 35]. Logistic likelihood, although convex, cannot however be inserted in a standard MIQP model. Still, [38] showed that, by means of a cutting-planes based approximation, a good surrogate MILP problem can be defined and solved, at least for moderate problem sizes, providing a high quality classification model.

The aim of this paper is to introduce a novel technique that, exploiting mixed-integer modeling, is able to produce good solutions to the best subset selection in logistic regression problem, being at the same time reasonably scalable w.r.t. problem size. To reach this goal, we make use of a decomposition strategy.

The main contributions of the paper consist in:

- the definition of a strong necessary optimality condition for optimization problems with an ℓ_0 penalty term;
- the definition of a decomposition scheme, with a suitable variable selection rule, allowing to improve the scalability of the method from [38], with convergence guarantees to points satisfying the aforementioned condition;
- practical suggestions to improve the performance of the proposed algorithm;
- a thorough computational study comparing various solvers from the literature on best subset selection problems in logistic regression.

The rest of the manuscript is organized as follows: in Sect. 2, we formally introduce the problem of best subset selection in logistic regression, state optimality conditions and provide a brief review of a related approach. In Sect. 3, we present our proposed method, explaining in detail the key contributions and carrying out a theoretical analysis of the procedure. Then, we describe and report in Sect. 4 the results of a thorough experimental comparison on a benchmark of real-world classification problems; these results highlight the effectiveness of the proposed approach with respect to state-of-the-art methods. We finally give some concluding remarks and suggest possible future research in Sect. 5. In Appendix we also provide a detailed review of the algorithms considered in the computational experiments.

2 Best subset selection in logistic regression

Let $X \in \mathbb{R}^{N\times n}$ be a dataset of N examples with n real features and $Y \in \{-1, 1\}^N$ a set of N binary labels. The *logistic regression model* [22] for binary classification defines the probability for an example x of belonging to class $y = 1$ as

$$\mathbb{P}(y = 1 \mid x) = \frac{1}{1 + \exp(-w^\top x)}.$$

Substantially, a sigmoid nonlinearity is applied to the output of a linear regression model. Note that the intercept term is not explicitly present in the linear part of the model; in fact, it can be implicitly added, by considering it as a feature which is equal to 1 in all examples; we did so in the experimental part of this work. It is easy to see that

$$\mathbb{P}(y = -1 \mid x) = 1 - \mathbb{P}(y = 1 \mid x) = \frac{1}{1 + \exp(w^\top x)}.$$

Hence, the logistic regression model can be expressed by the single equation here below:

$$\mathbb{P}(y \mid x) = \frac{1}{1 + \exp(-y w^\top x)}. \tag{1}$$

Under the hypothesis that the distribution of $(y \mid x)$ follows a Bernoulli distribution, we get that model (1) is associated with the following *log-likelihood* function:

$$\ell(w) = -\sum_{i=1}^{N} \log\left(1 + \exp\left(-y^{(i)} w^\top x^{(i)}\right)\right). \tag{2}$$

A function $f(v) = \log(1 + \exp(-v))$ is referred to as logistic loss function and is a convex function. The maximum likelihood estimation of (1), which requires the maximization of $\ell(w)$, is thus a convex continuous optimization problem.

Identifying a subset of features that provides a good trade-off between fit quality and model sparsity is a recurrent task in applications. Indeed, a sparse model might offer a better explanation of the underlying generating model; moreover, sparsity is statistically proved to improve the generalization capabilities of the model [44]; finally, a sparse model will be computationally more efficient.

Many different approaches have been proposed in the literature for the best subset selection problem which, we recall, is a specific form of model selection. Every model selection procedure has advantages and disadvantages as it is difficult to think that there might exist a single, correct, model for a specific application. Among the many different proposals, those which base subset selection on *information criteria* [12, 13, 28] stand out as the most frequently used, both for their computational appeal as well as for their deep statistical theoretical support. Information criteria are statistical tools to compare the quality of different models in terms of quality of fit and sparsity simultaneously. The two currently most popular information criteria are:

- the *Akaike Information Criterion* (AIC) [1, 2, 11]:

  $$\mathrm{AIC}(w) = -2\ell(w) + 2\|w\|_0;$$

  Comparing a set of candidate models, the one with smallest AIC is considered closer to the truth than the others. Since the log-likelihood, at its maximum point, is a biased upward estimator of the model selection target [12], the penalty term $2\|w\|_0$, i.e., the total number of parameters involved in the model, allows to correct this bias;
- the *Bayesian Information Criterion* (BIC) [39]:

  $$\mathrm{BIC}(w) = -2\ell(w) + \log(N)\|w\|_0;$$

  It has been shown [12, 28] that given a set of candidate models, the one which minimizes the BIC is optimal for the data, in the sense that it is the one that maximizes the marginal likelihood of the data under the Bayesian assumption that all candidate models have equal prior probabilities.

Although other models can be proposed for model selection, those based on the AIC and BIC, or their variant, are extremely popular thanks to their solid statistical properties.

In summary, when referred to logistic regression models, the problem of best subset selection based on information criteria like AIC or BIC has the form of the following optimization problem:

$$\min_{w \in \mathbb{R}^n} \mathcal{L}(w) + \lambda \|w\|_0, \tag{3}$$

where $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ is twice the negative log-likelihood of the logistic regression model ($\mathcal{L}(w) = -2\ell(w)$), which is a continuously differentiable convex function, $\lambda > 0$ is a constant depending on the choice of the information criterion and $\| \cdot \|_0$ denotes the ℓ_0 semi-norm of a vector. Given a solution $\bar{w}$, we will denote the set of its nonzero variables, also referred to as *support*, by $S(\bar{w}) \subseteq \{1, \dots, n\}$, while $\bar{S}(\bar{w}) = \{1, \dots, n\} \setminus S(\bar{w})$, denotes its complementary. In the following, we will also refer to the objective function as $\mathcal{F}(w) = \mathcal{L}(w) + \lambda \|w\|_0$.

Because of the discontinuous nature of the ℓ_0 semi-norm, solving problems of the form (3) is not an easy task. In fact, problems like (3) are well-known to be $\mathcal{NP}$-hard, hence, finding global minima is intrinsically difficult.

Lu and Zhang [32] have established necessary first-order optimality conditions for problem (3); in fact, they consider a more general, constrained version of the problem. In the unconstrained case we are interested in, such conditions reduce to the following.

Definition 1 A point $w^\star \in \mathbb{R}^n$ satisfies Lu–Zhang first order optimality conditions for problem (3) if $\nabla_j \mathcal{L}(w^\star) = 0$ for all $j \in \{1, \dots, n\}$ such that $w_j^\star \neq 0$.

As proved by [32], if $\mathcal{L}(w)$ is a convex function, as in the case of logistic regression log-likelihood, there is an equivalence relation between Lu–Zhang optimality and local optimality, meaning there exists a neighborhood V of $w^\star$ such that $\mathcal{F}(w^\star) \leq \mathcal{F}(w)$ for all $w \in V$.

Proposition 1 *Let $w^\star \in \mathbb{R}^n$. Then, $w^\star$ is a local minimizer for Problem* (3) *if and only if it satisfies Lu–Zhang first order optimality conditions.*

This may appear surprising at first glance. However, after a more careful thinking, it should be evident. Being $\mathcal{L}$ convex, a Lu–Zhang point is globally optimal w.r.t. the nonzero variables. As for the zero variables, since $\mathcal{L}$ is continuous, there exists a neighborhood such that the decrease in $\mathcal{L}$ is bounded by λ, which is the penalty term that is added to the overall objective function as soon as one of the zero variables is moved.

Unfortunately, the number of Lu–Zhang local minima is in the order of 2^n. Indeed, for any subset of variables, minimizing w.r.t. those components, while keeping fixed the others to zero, allows to obtain a point which satisfies Lu–Zhang conditions. Hence, satisfying the necessary and sufficient conditions of local optimality is indeed a quite weak feature in practice. On the other hand, being the search of an optimal subset of variables a well-known $\mathcal{NP}$-hard problem, requiring theoretical guarantees of global optimality is unreasonable. In conclusion, it should be clear that the evaluation and comparison of algorithms designed to deal with problem (3) have to be based on the quality of the solutions empirically obtained in experiments.

However, we can further characterize candidates for optimality by means of the following notion, which adapts the concept of CW-optimality for cardinality constrained problems defined by [4]. To this aim, we introduce the notation $w_{\neq i}$ to denote all the components of w except the i-th.

Definition 2 A point $w^\star \in \mathbb{R}^n$ is a CW-minimum for Problem (3) if

$$w_i^\star \in \underset{w_i}{\operatorname{argmin}}\ \mathcal{F}(w_i; w_{\neq i}^\star) \tag{4}$$

for all $i = 1, \dots, n$.

Equivalently, (4) could be expressed as

$$\begin{aligned} w^\star \in \underset{w}{\operatorname{argmin}}\ & \mathcal{F}(w) \\ \text{s.t.}\ & \|w - w^\star\|_0 \leq 1 \end{aligned} \tag{5}$$

CW-optimality is a stronger property than Lu–Zhang stationarity. We outline this fact in the following proposition.

Proposition 2 *Consider Problem* (3) *and let* $w^\star \in \mathbb{R}^n$. *The following statements hold:*

1. *If* $w^\star$ *is a CW-minimum for* (3), *then* $w^\star$ *satisfies Lu–Zhang optimality conditions, i.e.,* $w^\star$ *is a local minimizer for* $w^\star$.
2. *If* $w^\star$ *is a global minimizer for* (3), *then* $w^\star$ *is a CW-minimum for* (3).

Proof We prove the statements one at a time.

1. Let $w^\star$ be a CW-minimum, i.e.,

$$w_i^\star \in \underset{w_i}{\operatorname{argmin}}\ \mathcal{F}(w_i; w_{\neq i}^\star) \tag{6}$$

for all $i = 1, \dots, n$. Assume by contradiction that $w^\star$ does not satisfy Lu–Zhang conditions; then, there exists $h \in \{1, \dots, n\}$ such that $w_h^\star \neq 0$ and $\nabla_h \mathcal{L}(w^\star) > 0$. Hence, $-\nabla_h \mathcal{L}(w^\star)$ is a descent direction for $\mathcal{L}(w_h; w_{h\neq j}^\star)$ at $w_h^\star \neq 0$, which contradicts (6).

2. Let $w^\star$ be a globally optimal point for (3). Assume by contradiction that $w^\star$ is not a CW-minimum, i.e., there exists $h \in \{1. \dots, n\}$ such that there exists $\hat{w}_h$ such that $\mathcal{F}(\hat{w}_h; w^\star_{h \neq j}) < \mathcal{F}(w^\star)$. This clearly contradicts that $w^\star$ is a global optimum.

□

Note that CW-optimality is a sufficient, yet not necessary, condition for local optimality. Indeed, Lu–Zhang conditions, and hence local optimality, certify that an improvement cannot be achieved without changing the set of nonzero variables. CW-optimality allows to also take into account possible changes in the support, although limited to one variable. We show this in the following examples, where, for the sake of simplicity, we consider a simpler convex function than $\mathcal{L}$.

Example 1 Consider the problem

$$\min_{w \in \mathbb{R}^2} \varphi(w) = (w_1 - 1)^2 + (w_2 - 2)^2 + 2\|w\|_0.$$

It is easy to see that Lu–Zhang conditions are satisfied by the points $w^a = (0, 0)$, $w^b = (1, 2)$, $w^c = (0, 2)$ and $w^d = (1, 0)$. We have $\varphi(w^a) = 5$, $\varphi(w^b) = 4$, $\varphi(w^c) = 3$, $\varphi(w^d) = 4$. We can then observe that w^c and w^d are CW-minima, as their objective value cannot be improved by changing only one of their components, while w^a and w^b are not CW-optima, as the solutions can be improved by zeroing a component or setting the first component to 1, respectively.

We can conclude by remarking that searching through the CW-points allows to filter out a number of local minima that are certainly not globally optimal.

2.1 The MILO approach

Many approaches have been proposed to tackle cardinality-penalized problems in general and for problem (3) specifically. We provide a detailed review of many of these methods in Appendix. Here, we focus on a particular approach that is relevant for the rest of the paper.

Sato et al. [38] proposed a mixed integer linear (MILO) reformulation for problem (3), which is, to the best of our knowledge, the top performing one, as long as the dimensions of the underlying classification problem are not exceedingly large. Such approach has two core ideas. The first one consists of the replacement of the ℓ_0 term by the sum of binary indicator variables.

The second key element is the approximation of the nonlinearity in $\mathcal{L}$, i.e., the logistic loss function, by a piecewise linear function, so that the resulting reformulated problem is a MILP problem. The approximating piecewise linear function is defined by the pointwise maximum of a family of tangent lines, that is,

$$\begin{aligned} f(v) = \log(1 + \exp(-v)) \approx \hat{f}(v) &= \max\{f'(v^k)(v - v^k) + f(v^k) \mid k = 1, 2, \dots, K\} \\ &= \min\{t \mid t \geq f'(v^k)(v - v^k) + f(v^k),\ k = 1, \dots, K\} \end{aligned}$$

for some discrete set of points $\{v^1, \dots, v^K\}$. The function $\hat{f}$ is a linear underestimator to the true loss logistic function. The final MILP reformulation of problem (3) is given by

$$\begin{aligned}
&\min_{w,z,t} 2\sum_{i=1}^{N} t_i + \lambda \sum_{i=1}^{n} z_i \\
&\text{s.t. } -Mz_i \le w_i \le Mz_i \quad \forall\, i = 1, \dots, n, \\
&\qquad z \in \{0,1\}^n, \\
&\qquad t_i \ge f'(v^k)(y^{(i)}(w^\top x^{(i)}) - v^k) + f(v^k) \quad \forall\, k = 1, \dots, K, \quad \forall\, i = 1, \dots, N,
\end{aligned} \tag{7}$$

where M is a large enough positive constant.

The choice of the tangent lines is clearly crucial for this method. For large values of K, problem (7) becomes hard to solve. On the other hand, if the number of lines is small, the quality of the approximation will reasonably be low. Hence, points v^k should be selected carefully. Sato et al. [38] suggest to adopt a greedy algorithm that adds one tangent line at a time, minimizing the area of gap between the exact logistic loss and the linear piece-wise approximation. In their work, Sato et al. [38] show that the greedy algorithm provides, depending on the desired set size, the following sets of interpolation points:

$$\begin{aligned}
&V_1 = \{0, \pm 1.9, \pm\infty\}, \qquad V_2 = V_1 \cup \{\pm 0.89, \pm 3.55\}, \\
&V_3 = V_2 \cup \{\pm 0.44, \pm 1.37, \pm 2.63, \pm 5.16\}
\end{aligned}$$

As problem (7) employs an approximation of $\mathcal{L}$, the optimal solution $\hat{w}$ obtained by solving it is not necessarily optimal for (3). However, since the objective of (7) is an underestimator of the original objective function, it is possible to make a posteriori accuracy evaluations. In particular, letting $w^\star$ be the optimal solution and

$$\hat{\mathcal{L}}(w) = 2\sum_{i=1}^{N} \max_k f'(v^k)(y^{(i)}(w^\top x^{(i)}) - v^k) + f(v^k),$$

we have

$$\hat{\mathcal{L}}(\hat{w}) + \lambda\|\hat{w}\|_0 \le \mathcal{L}(w^\star) + \|w^\star\|_0 \le \mathcal{L}(\hat{w}) + \lambda\|\hat{w}\|_0.$$

Hence, if $\mathcal{L}(\hat{w}) - \hat{\mathcal{L}}(\hat{w})$ is small, it is guaranteed that the value of the real objective function at $\hat{w}$ is close to the optimum.

3 The proposed method

The MILO approach from [38] is computationally very effective, but it suffers from a main drawback: it scales pretty badly as either the number of examples or the number of features in the dataset grows. This fact is also highlighted by the experimental results reported in the original MILO paper.

On the other hand, heuristic enumerative-like approaches present the limitation of performing moves with a limited horizon. This holds not only for the simple stepwise procedures, but also for other possible more complex and structured strategies that one may come up with. Indeed, selecting one move among all those involving the addition or removal from the current best subset of multiple variables at one time is unsustainable except for tiny datasets.

In this work, we propose a new approach that somehow employs the MILO formulation to overcome the limitations of discrete enumeration methods, but also has better scalability features than the standard MILO approach itself, in particular w.r.t. the number of features. The core idea of our proposal consists of the application of a decomposition strategy to problem (3). The classical *Block Coordinate Descent* (BCD) [5, 42] algorithm consists in performing, at each iteration, the optimization w.r.t. one block of variables, i.e., the iterations have the form

$$w_{B_\ell}^{\ell+1} \in \underset{w_{B_\ell}}{\operatorname{argmin}}\ \mathcal{F}(w_{B_\ell}; w_{\bar{B}_\ell}^{\ell}), \tag{8}$$

$$w_{\bar{B}_\ell}^{\ell+1} = w_{\bar{B}_\ell}^{\ell}, \tag{9}$$

where $B_\ell \subset \{1, \dots, n\}$ is referred to as *working set*, $\bar{B}_\ell = \{1, \dots, n\} \setminus B_\ell$. Now, if the working set size $|B|$ is reasonably small, the subproblems can be easily handled by means of a MILO model analogous to that from [38]. Carrying out such a strategy, the subproblems to be solved at each iteration have the form

$$\begin{aligned} \min_{w_{B_\ell}, z, t}\ & 2\sum_{i=1}^{N} t_i + \lambda \sum_{i \in B_\ell} z_i \\ \text{s.t.}\ & -Mz_i \le w_i \le Mz_i \quad \forall\, i \in B_\ell, \\ & z_i \in \{0, 1\} \quad \forall\, i \in B_\ell, \\ & t_i \ge f'(v^k)(y_i(w^\top x_i) - v^k) + f(v^k) \quad \forall\, k = 1, \dots, K, \quad \forall\, i = 1, \dots, N. \\ & w_{\bar{B}_\ell} = w_{\bar{B}_\ell}^{\ell} \end{aligned} \tag{10}$$

At the end of each iteration, we can also introduce a minimization step of $\mathcal{L}$ w.r.t. the current nonzero variables. Since this is a convex minimization step, it allows to refine every iterate up to global optimality w.r.t. the support and to Lu–Zhang stationarity, i.e., local optimality, in terms of the original problem. This operation has low computational cost and a great practical utility, since it guarantees, as we will show in the following, finite termination of the algorithm.

3.1 The working set selection rule

Many different strategies could be designed for selecting, at each iteration ℓ, the variables constituting the working set B_ℓ, within the BCD framework. In this work, we propose a rule based on the violation of CW-optimality.

Given the current iterate x^ℓ, we can define a score function

$$p(w^\ell, i) = \begin{cases} \mathcal{L}(0, w^\ell_{\neq i}) - \lambda + \lambda\|w^\ell\|_0 & \text{if } w^\ell_i \neq 0, \\ \min_{w_i} \mathcal{L}(w_i, w^\ell_{\neq i}) + \lambda + \lambda\|w^\ell\|_0 & \text{if } w^\ell_i = 0. \end{cases} \tag{11}$$

The rational of this score is to estimate what the objective function would become if we forced the considered variable w_i alone to change its status, entering/leaving the support.

We finally select the working set B^ℓ, of size b, choosing, in a greedy way, the b lowest scoring variables, i.e., by solving the problem

$$\begin{aligned} B^\ell \in \arg\ \min\ _B & \sum_{h \in B} p(w^\ell, h) \\ \text{s.t. } & B \subseteq \{1, \dots, n\}, \\ & |B| = b. \end{aligned} \tag{12}$$

3.2 The complete procedure

The whole proposed algorithm is formally summarized in Algorithm 1. Basically, it is a BCD where subproblems are (approximately) solved by the MILO reformulation and variables are selected by (12).

In addition, there are some technical steps aimed at making the algorithm work from both the theoretical and the practical point of view.

In the ideal case where the subproblems are solved exactly, thanks to our selection rule, we would be guaranteed to do at least as well as a greedy descent step along a single variable. However, subproblems are approximated and it happens that, solving the MILO, the true objective may sometimes not be decreased, even if the simple greedy step would. In such cases, we actually perform the greedy step to produce the next iterate.

Moreover, at the end of each iteration we perform the refinement step previously discussed. Note that this step cannot increase the value of $\mathcal{F}$, as we are lowering the value of $\mathcal{L}$ by only moving nonzero variables.

Last, we make the stopping criterion explicit; the algorithm stops as soon as an iteration is not able to produce a decrease in the objective value; we then return the point w^ℓ.

Algorithm 1: MILO-BCD

```
Input: $w^0 = 0,\ b < n$.
for $\ell = 0, 1, \dots$ do
    Select the working set $B^\ell$ using rule (12)
    Compute $\nu^{\ell+1}_{B^\ell}$ by solving problem (10).
    Set $\nu^{\ell+1}_{\bar{B}^\ell} = w^\ell_{\bar{B}^\ell}$
    if $\mathcal{F}(\nu^{\ell+1}) \geq \mathcal{F}(w^\ell)$ then
        Set
            $\nu^{\ell+1} \in \arg\min_{w} \mathcal{F}(w)$
            $\quad$ s.t. $\|w^\ell - w\|_0 \leq 1$
            $\quad\quad w_{\bar{B}^\ell} = w^\ell_{\bar{B}^\ell}$
    Set
        $w^{\ell+1} \in \arg\min_{w} \mathcal{L}(w)$
        $\quad$ s.t. $\quad w_i = 0$ for all $i \in \bar{S}(\nu^{\ell+1})$
    if $\mathcal{F}(w^{\ell+1}) = \mathcal{F}(w^\ell)$ then
        return $w^\ell$
```

3.3 Theoretical analysis

In this section, we provide a theoretical characterization for Algorithm 1.

We begin by stating a nice property of the set of local minima of problem (3).

Lemma 1 *Let* $\Gamma = \{\mathcal{F}(w) \mid w \text{ is a local minimum point for problem (3)}\}$. *Then* $|\Gamma| \leq 2^n$.

Proof For each support set $S \subseteq \{1, \dots, n\}$ let $L^\star_S$ be the optimal value of the problem

$$\min_{w:\, w_{\bar{S}}=0} \mathcal{L}(w).$$

Let $w^\star$ be a local minimizer for problem (3). Then, from Lu–Zhang conditions and the convexity of $\mathcal{L}$, it is a global minimizer of

$$\min_{w:\, w_{\bar{S}(w^\star)}=0} \mathcal{L}(w),$$

and $\mathcal{F}(w^\star) = L^\star_{S(w^\star)} + \lambda |S(w^\star)|$. We hence have

$$\begin{aligned} \Gamma &= \{L^\star_{S(w^\star)} + \lambda |S(w^\star)| \mid w^\star \text{ is a local minimizer for (3)}\} \\ &\subseteq \{L^\star_S + \lambda |S| \mid S \subseteq \{1, \dots, n\}\} \end{aligned}$$

and so

$$|\Gamma| \le |\{L^\star_S + \lambda|S| \mid S \subseteq \{1,\dots,n\}\}| \le |\{S \mid S \subseteq \{1,\dots,n\}\}| = 2^n.$$

□

We go on with a statement about the relationship between the objective function $\mathcal{F}(w)$ and the score function $p(w, i)$.

Lemma 2 *Let p be the score function defined as in (11) and let $\bar{w} \in \mathbb{R}^n$. Moreover, for all $h = 1,\dots,n$, let $\bar{w}^h \in \operatorname{argmin}_{w_h} \mathcal{F}(w_h, \bar{w}_{\neq h})$. Then the following statements hold*

(1) *If $\mathcal{F}(\bar{w}^h) = \mathcal{F}(\bar{w})$ then $p(\bar{w}, h) \ge \mathcal{F}(\bar{w})$;*
(2) *If $\mathcal{F}(\bar{w}^h) < \mathcal{F}(\bar{w})$ and $\bar{w}$ satisfies Lu–Zhang conditions, then $p(\bar{w}, h) = \mathcal{F}(\bar{w}^h)$.*

Proof We prove the two statements one at a time:

(i) Let us assume that the thesis is false, i.e., $\mathcal{F}(\bar{w}^h) = \mathcal{F}(\bar{w})$ and $p(\bar{w}, h) < \mathcal{F}(\bar{w})$. We distinguish two cases: $\bar{w}_h = 0$ and $\bar{w}_h \neq 0$. In the former case we have

$$\begin{aligned}\mathcal{F}(\bar{w}) > p(\bar{w}, h) &= \min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda + \lambda\|\bar{w}\|_0 \\ &= \min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda + \lambda\|\bar{w}_{\neq h}\|_0 \\ &\ge \min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda\|w_h\|_0 + \lambda\|\bar{w}_{\neq h}\|_0 \\ &= \min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda\|(w_h, \bar{w}_{\neq h})\|_0 \\ &= \mathcal{F}(\bar{w}^h) = \mathcal{F}(\bar{w}),\end{aligned}$$

which is absurd. In the latter case, we have

$$\begin{aligned}\mathcal{F}(\bar{w}) > p(\bar{w}, h) &= \mathcal{L}(0, \bar{w}_{\neq h}) - \lambda + \lambda\|\bar{w}\|_0 \\ &= \mathcal{L}(0, \bar{w}_{\neq h}) + \lambda\|(0, \bar{w}_{\neq h})\|_0 \\ &\ge \mathcal{F}(\bar{w}^h) = \mathcal{F}(\bar{w}),\end{aligned}$$

which is again a contradiction; hence we get the thesis.

(ii) We again distinguish two cases: $\bar{w}_h = 0$ and $\bar{w}_h \neq 0$. In the first case we have

$$\begin{aligned}\mathcal{F}(\bar{w}^h) &= \min_{w_h} \mathcal{F}(w_h, \bar{w}_{\neq h}) \\ &= \min\left\{\min_{w_h \neq 0} \mathcal{F}(w_h, \bar{w}_{\neq h}), \mathcal{F}(0, \bar{w}_{\neq h})\right\} \\ &= \min\left\{\min_{w_h \neq 0} \mathcal{F}(w_h, \bar{w}_{\neq h}), \mathcal{F}(\bar{w})\right\}\end{aligned}$$

But since we know $\mathcal{F}(\bar{w}^h) < \mathcal{F}(\bar{w})$, we can imply that

$$\min_{w_h \neq 0} \mathcal{L}(w_h, \bar{w}_{\neq h}) < \mathcal{L}(0, \bar{w}_{\neq h})$$

and we can also write

$$\begin{aligned}
\mathcal{F}(\bar{w}^h) &= \min_{w_h \neq 0} \mathcal{F}(w_h, \bar{w}_{\neq h}) \\
&= \min_{w_h \neq 0} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda \|(w_h, \bar{w}_{\neq h})\|_0 \\
&= \min_{w_h \neq 0} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda + \lambda \|\bar{w}_{\neq h}\|_0 \\
&= \min_{w_h \neq 0} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda + \lambda \|\bar{w}\|_0 \\
&= \min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda + \lambda \|\bar{w}\|_0 \\
&= p(\bar{x}, h).
\end{aligned}$$

In the second case, since $\bar{w}$ satisfies Lu–Zhang conditions, we have $\bar{w}_h \in \arg\min_{w_h} \mathcal{L}(w_h, \bar{w}_{\neq h})$. Therefore

$$\bar{w}_h \in \arg\min_{w_h \neq 0} \mathcal{L}(w_h, \bar{w}_{\neq h}) + \lambda \|(w_h, \bar{w}_{\neq h})\|_0 = \arg\min_{w_h \neq 0} \mathcal{F}(w_h, \bar{w}_{\neq h}).$$

Since $\mathcal{F}(\bar{w}^h) < \mathcal{F}(\bar{w}) = \min_{w_h \neq 0} \mathcal{F}(w_h, \bar{w}_{\neq h})$, we get $\bar{w}^h = (0, \bar{w}_{\neq h})$. We finally obtain

$$\begin{aligned}
\mathcal{F}(\bar{w}^h) &= \mathcal{L}(\bar{w}^h) + \lambda \|\bar{w}^h\|_0 \\
&= \mathcal{L}(0, \bar{w}_{\neq h}) + \lambda \|(0, \bar{w}_{\neq h})\|_0 \\
&= \mathcal{L}(0, \bar{w}_{\neq h}) + \lambda \|\bar{w}_{\neq h}\|_0 \\
&= \mathcal{L}(0, \bar{w}_{\neq h}) + \lambda \|\bar{w}\|_0 - \lambda \\
&= p(\bar{w}, h).
\end{aligned}$$

□

We are finally able to state finite termination and optimality properties of the returned solution of the MILO-BCD procedure.

Proposition 3 *Let $\{w^\ell\}$ be the sequence generated by Algorithm 1. Then $\{w^\ell\}$ is a finite sequence and the last element $\bar{w}$ is a CW-minimum for problem* (3).

Proof From the instructions of the algorithm, for all $\ell = 1, 2, \ldots$, we have that

$$\begin{aligned}
w^\ell \in \underset{w}{\operatorname{argmin}} \quad & \mathcal{L}(w) \\
\text{s.t.} \quad & w_i = 0 \text{ for all } i \in \bar{S}(w^\ell),
\end{aligned}$$

hence $\nabla_i \mathcal{L}(w^\ell) = 0$ for all $i \in S(w^\ell)$, i.e., w^ℓ satisfies Lu–Zhang conditions and is therefore a local minimum point for problem (3). From Lemma 1, we thus know that there exist finite possible values for $\mathcal{F}(w^\ell)$. Moreover, $\{\mathcal{F}(w^\ell)\}$ is a nonincreasing sequence. We can conclude that in a finite number of iterations we get $\mathcal{F}(w^\ell) = \mathcal{F}(w^{\ell+1})$, activating the stopping criterion.

We now prove that the returned point, $\bar{w} = w^{\bar{\ell}}$ for some $\bar{\ell} \in \mathbb{N}$, is CW-optimal. Assume by contradiction that $\bar{w}$ is not CW-optimal. Then, there exists $h \in \{1, \dots, n\}$ such that $\min_{w_h} \mathcal{F}(w_h, \bar{w}_{\neq h}) < \mathcal{F}(\bar{w})$.

We show that this implies that there exists $t \in \{1, \dots, n\}$ such that $t \in B^{\bar{\ell}}$ and $\min_{w_t} \mathcal{F}(w_t, \bar{w}_{\neq t}) < \mathcal{F}(\bar{w})$. Assume by contradiction that for all $j \in B^{\bar{\ell}}$ it holds $\min_{w_j} \mathcal{F}(w_j, \bar{w}_{\neq j}) = \mathcal{F}(\bar{w})$. Letting i any index in the working set $B^{\bar{\ell}}$ and recalling Lemma 2, we have

$$\begin{aligned}
\sum_{j \in B^{\bar{\ell}}} p(w^{\bar{\ell}}, j) &= \sum_{j \in B^{\bar{\ell}} \setminus \{i\}} p(w^{\bar{\ell}}, j) + p(w^{\bar{\ell}}, i) \\
&\geq \sum_{j \in B^{\bar{\ell}} \setminus \{i\}} p(w^{\bar{\ell}}, j) + \mathcal{F}(w^{\bar{\ell}}) \\
&> \sum_{j \in B^{\bar{\ell}} \setminus \{i\}} p(w^{\bar{\ell}}, j) + p(w^{\bar{\ell}}, h) \\
&= \sum_{j \in B^{\bar{\ell}} \cup \{h\} \setminus \{i\}} p(w^{\bar{\ell}}, j),
\end{aligned}$$

which contradicts the working set selection rule (12).

Now, either $\mathcal{F}(v^{\ell+1}) < \mathcal{F}(w^{\bar{\ell}})$ after steps 4–5 of the algorithm, or, after step 7, we get

$$\mathcal{F}(v^{\ell+1}) \leq \min_{w_t} \mathcal{F}(w_t, w^{\bar{\ell}}_{\neq t}) < \mathcal{F}(w^{\bar{\ell}}).$$

Therefore, since step 8 cannot increase the value of $\mathcal{F}$, we get $\mathcal{F}(w^{\bar{\ell}+1}) < \mathcal{F}^{\bar{\ell}}$, but this contradicts the fact that the stopping criterion at line 9 is satisfied at iteration $\bar{\ell}$.

□

3.4 Finding good CW-optima

We have shown in the previous section that Algorithm 1 always returns a CW-optimal solution. Although this allows us to cut off a lot of local minima, there are in practice many low-quality CW-minima. For this reason, we introduce in our algorithm an heuristic aimed at leaving bad CW-optima where it may get stuck.

In detail, we do as follows. Instead of stopping the algorithm as soon as the objective value does not decrease, we try to repeat the iteration with a different working set. In doing this, we obviously have to change the working set selection rule. This operation is repeated up to a maximum number of times. If after testing a suitable amount of different working sets a decrease in the objective function cannot be achieved, the algorithm stops.

Specifically, we define a modified score function

$$\hat{p}(w^{\ell}, i) = p(w^{\ell}, i) + 2^{r_i} - 1, \tag{13}$$

where r_i is the number of times the i-th variable was in the working set in the previous attempts.

The idea of this working set selection rule is to first try a greedy selection. Then, if that first attempt failed, we penalize (exponentially) variables that were tried more times and could not provide improvements in the end. This penalty is heuristic. In fact, we may end up with repeating the search over the same working set from the same starting point. However, we can keep track of the working set used throughout the outer iteration, in order to avoid duplicate computations.

Note that such a modification does not alter the theoretical properties of the algorithm; on the other hand, it has a massive impact on the empirical performance.

4 Computational results

This section is dedicated to a computational comparison between the approach proposed in this paper and the state-of-the-art algorithms described in Sect. 2 and Appendix. In our experiments we took into account 11 datasets for binary classification tasks, listed in Table 1, from the UCI Machine Learning Repository [15]. In fact, the `digits` dataset is inherently for multi-class classification; we followed the same binarization strategy as [38], assigning a positive label to the examples from the largest class and a negative one to all the others. Moreover, we removed data points with missing variables, encoded the categorical variables with a one-hot vector and normalized the other ones to zero mean and unit variance. In Table 1 we also reported the number n of data points and the number p of features of each dataset, after the aforementioned preprocessing operations.

These datasets constitute a benchmark to evaluate the performance of the algorithms under examination, namely: Forward Selection and Backward Elimination Stepwise heuristics, LASSO, Penalty Decomposition, Concave approximation, the Outer Approximation method in its original form, in the adapted version for cardinality-penalized problems and also in the variant exploiting the

Table 1 List of datasets used for the experiments on best subset selection in logistic regression

Dataset	n	p	Abbreviation
Parkinsons	195	22	`parkinsons`
Heart (statlog)	270	25	`heart`
Breast cancer wisconsin (prognostic)	194	33	`breast`
QSAR biodegradation	1055	41	`biodeg`
SPECTF heart	267	44	`spectf`
Spambase	4601	57	`spam`
Optical recognition of handwritten digits	3823	62	`digits`
Libras movement	360	90	`libras`
a2a	2265	123	`a2a`
w2a	2470	300	`w2a`
Madelon	2000	500	`madelon`

approximated dual problems, MILO and our proposed method MILO-BCD. All of these algorithms are described in Appendix and Sect. 2.

All the experiments described in this section were performed on a machine with Ubuntu Server 18.04 LTS OS, Intel Xeon E5-2430 v2 @ 2.50 GHz CPU and 16GB RAM. The algorithms were implemented in Python 3.7.4, exploiting Gurobi 9.0.0 [20] for the outer approximation method, MILO and MILO-BCD. The `scipy` [43] implementation of the L-BFGS algorithm defined in [30] was employed for local optimization steps of all methods. A time limit of 10,000 s was set for each method.

Both the stepwise methods (forward and backward) exploit L-BFGS [30] as internal optimizer. The forward selection version uses L-BFGS to optimize the logistic with respect to one variable, whereas backward elimination defines his starting point exploiting L-BFGS to optimize the model w.r.t. all the variables.

Concerning LASSO, we solved Problem (14) using the `scikit-learn` implementation [36], with LIBLINEAR library [18] as internal optimizer, for each value of the hyperparameter λ. Each λ value was chosen so that two different hyperparameters, $\lambda_1 \neq \lambda_2$, would not produce the same level of sparsity and to avoid the zero solution. More specifically, we defined our set of hyperparameters by computing the LASSO path, exploiting to the `scikit-learn` function `l1_min_c`. All the obtained solutions were refined by further optimizing w.r.t. the nonzero components only by means of L-BFGS. At the end of this grid search we selected the solution, among these one, providing the best information criterion value.

Penalty Decomposition requires to set a large number of hyperparameters: in our experiments we set $\epsilon = 10^{-1}$, $\eta = 10^{-3}$ and $\sigma_\epsilon = 1$ for all the datasets. We ran the algorithm multiple times for values of τ and σ_τ taken from a small grid. L-BFGS was again used as internal solver. The best solution obtained, in terms of information criterion, was retained at the end of the process.

Concave approximation, theoretically, requires the solution of a sequence of problem. However, as outlined in Appendix, a single problem with fixed approximation hyperparameter μ can be solved in practice [37]. In our experiments, Problem (17) was solved by using L-BFGS. Again, we retain as optimal solution the one that, after an L-BFGS refining step w.r.t. the nonzero variables, minimizes the information criterion among a set of resulting points obtained for different values of μ.

It is important to highlight that the refining optimization step is crucial for methods like the Concave Approximation or LASSO; as a matter of fact, without this precaution, the computed solutions don't even necessarily satisfy the Lu–Zhang conditions.

All variants of the Outer Approximation method expoit Gurobi to handle the MILP subproblems and L-BFGS for the continuous ones. As suggested by [8], a single branch and bound tree is constructed to solve all the MILP subproblems, adding cutting-type constraints dynamically as lazy constraints. Moreover, the starting cut is decided by means of the first-order heuristic described in the referenced work. For the cardinality-constrained version of the algorithm, we set a time limit of 1000 s for the solution of any individual problem of the form (18) with a fixed value of s. As for the dual formulation, we set $\gamma = 10^4$ to make the considered problem as close as possible to the formulation tackled by all other algorithms. The approximated version of the dual problem, which is quadratic, is efficiently solved with Gurobi instead of L-BFGS.

As concerns MILO and MILO-BCD, we employed the V_2 set of interpolation points for both methods, in order to have a good trade-off between accuracy and computational burden. Moreover, for MILO-BCD we set the cardinality of the working set b to 20 for all the problems. We report in Sect. 4.1 the results of preliminar computational experiments that appear to support our choice. All the subproblems were solved with Gurobi. For MILO-BCD we employ the heuristic discussed in Sect. 3.4. For each problem, the maximum number of consecutive attempts with no improvement, before stopping the algorithm, is set to n. Note that, in order to improve the algorithm efficiency, we instantiate a single MILP problem with n variables and dynamically change the box constraints based on the current working set. The continuous optimization steps needed to perform steps 7 and 8 of Algorithm 1 are performed by using L-BFGS.

In Tables 2, 3, 4 and 5 the computational results of minimizing AIC and BIC respectively on the 11 datasets are shown. For each algorithm and problem, we can see the information criterion value at the returned solution, its zero norm and the total runtime. We can observe the effectiveness of the MILO-BCD approach w.r.t. the other methods. In particular in 8 out of 11 test problems MILO-BCD found the best AIC value, while in the remaining three cases it attains a very close second-best result. The results of minimizing BIC are very similar: for 9 out of 11 datasets MILO-BCD returns the best solution and in the remaining two it ranks at the second place. We can also note that, in cases where p is large such as `spam`, `digits`, `a2a`, `w2a` and `madelon` datasets, our method, within the established time limit, is able to find a much better quality solution with respect to the other algorithms (with the only exception of `spam` for the AIC), and in particular compared to MILO.

As for the efficiency, Tables 2, 3, 4 and 5 also allow to evaluate the computational burden of MILO-BCD. As expected, our method is slower than the approaches that are not based on Mixed Integer Optimization, which on the other hand provide lower quality solutions. However, compared to standard MILO, we can see a considerable improvement in terms of computational time with both the small and the large datasets.

In Fig. 1 we plot the cumulative distribution of absolute distance from the optimum attained by each solver, computed upon the 22 subset selection problems. The x-axis values represent the difference in absolute value between the information criterion obtained and the best one found, while y-axis reports the fraction of solved problems within a certain distance from the best. As it is possible to see, MILO-BCD clearly outperforms the other methods. As a matter of fact, MILO-BCD always found a solution that is distant less than 15 from the optimal one and in around the 80% of the problems it attained the optimal solution. We can also see that for all the other methods there is a number of bad cases where the obtained value is very far from the optimal one. Note that we consider the absolute distance from the best, instead of a relative distance, since it is usually the difference in IC values which is considered in practice to assess the quality of a model w.r.t. another one [12].

Finally, we highlight that MILO-BCD manages to greatly increase the performance of MILO, without making its interface more complex. As a matter of fact,

Table 2 Results of AIC minimization in logistic regression with different optimization methods on small datasets (best result for each dataset in bold)

Dataset	Method	AIC	ℓ_0	Time (s)
parkinsons	Forward stepwise	129.2567	5	0.280
	Backward stepwise	126.6948	17	0.172
	LASSO	129.7412	16	6.290
	Penalty decomposition	134.1499	2	22.486
	Concave approximation	129.6589	17	2.899
	Outer approximation CC	115.8998	9	≥ 10,000
	Outer approximation CP	120.6478	11	≥ 10,000
	Outer approximation dual	128.1812	17	3.278
	MILO	113.5371	8	12.531
	MILO-BCD	**113.5005**	8	93.708
heart	Forward stepwise	197.6972	11	0.577
	Backward stepwise	216.6682	23	0.038
	LASSO	202.4335	15	2.282
	Penalty decomposition	226.1013	4	49.500
	Concave approximation	206.4321	17	1.384
	Outer approximation CC	206.8911	12	≥ 10,000
	Outer approximation CP	263.2117	5	≥ 10,000
	Outer approximation dual	207.2493	19	4.087
	MILO	195.7757	11	41.593
	MILO-BCD	**195.6242**	11	95.399
breast	Forward stepwise	180.4932	6	0.470
	Backward stepwise	163.2610	33	0.413
	LASSO	156.6797	24	21.321
	Penalty decomposition	189.2942	2	8.044
	Concave approximation	158.11729	24	3.398
	Outer approximation CC	166.1055	34	≥ 10,000
	Outer approximation CP	202.8904	9	≥ 10,000
	Outer approximation dual	161.6405	31	6.675
	MILO	**147.5119**	19	86.250
	MILO-BCD	147.6781	17	236.126
biodeg	Forward stepwise	703.9588	20	3.582
	Backward stepwise	661.6047	32	0.417
	LASSO	665.1640	32	65.344
	Penalty decomposition	671.8854	18	232.120
	Concave approximation	663.5171	24	5.789
	Outer approximation CC	678.4316	42	≥ 10,000
	Outer approximation CP	1263.0706	6	≥ 10,000
	Outer approximation dual	681.6687	31	29.329
	MILO	**653.4768**	23	6885.277
	MILO-BCD	654.4053	25	707.356

Table 2 (continued)

Dataset	Method	AIC	ℓ_0	Time (s)
spectf	Forward stepwise	178.9840	6	0.797
	Backward stepwise	180.0595	28	0.214
	LASSO	181.4678	13	8.966
	Penalty decomposition	222.8672	2	55.287
	Concave approximation	181.8271	17	3.788
	Outer approximation CC	178.8349	12	≥ 10,000
	Outer approximation CP	222.3555	5	≥ 10,000
	Outer approximation dual	206.1484	38	10.766
	MILO	168.5162	15	1293.650
	MILO-BCD	**168.3443**	15	205.6255
libras	Forward stepwise	53.3215	11	2.558
	Backward stepwise	152.0723	76	0.470
	LASSO	28.0720	14	5.413
	Penalty decomposition	142.4580	2	530.951
	Concave approximation	70.0072	35	12.532
	Outer approximation CC	33.4904	11	≥ 10,000
	Outer approximation CP	72.4350	6	≥ 10,000
	Outer approximation dual	64.0018	32	58.061
	MILO	14.2040	7	≥ 10,000
	MILO-BCD	**14.1557**	7	654.227

we have only added a hyperparameter that controls the cardinality of the working set and experimentally appears to be extremely easy to tune. Indeed, note that all the experiments were carried out using the same working set size for each dataset and, despite this choice, MILO-BCD shown impressive performances in all the considered datasets.

4.1 Varying the working set size

The value of the working set size *b* may greatly affect the performance of the MILO-BCD procedure, in terms of both quality of solutions and running time. For this reason, we performed a study to evaluate the behavior of the algorithm as the value of *b* changes. We ran MILO-BCD on the problems obtained from datasets at different scales: `heart`, `breast`, `spectf`, and `a2a`. AIC is used as GOF measure.

The results are reported in Table 6 and Fig. 2. We can see that a working set size of 20, as employed in the experiments of the previous section, provides a good trade-off. Indeed, the running time seems to grow in general with the working set size, whereas the optimal solution is approached only when large working

Table 3 Results of AIC minimization in logistic regression with different optimization methods on large datasets (best result for each dataset in bold)

Dataset	Method	AIC	ℓ_0	Time (s)
spam	Forward stepwise	1906.5143	45	36.136
	Backward stepwise	1901.9650	46	1.468
	LASSO	**1892.6580**	53	1209.108
	Penalty decomposition	5244.4292	3	≥ 10,000
	Concave approximation	1916.1159	51	13.654
	Outer approximation CC	1963.0467	52	≥ 10,000
	Outer approximation CP	2138.2306	36	≥ 10,000
	Outer approximation dual	1931.7670	58	161.062
	MILO	1909.0709	44	≥ 10,000
	MILO-BCD	1904.2989	44	8442.004
digits	Forward stepwise	378.6893	25	13.139
	Backward stepwise	341.8344	42	1.894
	LASSO	346.9967	43	2154.283
	Penalty decomposition	7168.3316	1	≥ 10,000
	Concave approximation	338.1436	31	24.398
	Outer approximation CC	386.4583	64	≥ 10,000
	Outer approximation CP	686.1014	12	≥ 10,000
	Outer approximation dual	372.3541	44	125.282
	MILO	323.6231	26	≥ 10000
	MILO-BCD	**322.7531**	25	6557.441
a2a	Forward stepwise	1605.9851	34	20.864
	Backward stepwise	1659.0279	87	4.038
	LASSO	1615.6245	60	394.008
	Penalty decomposition	1676.8714	16	605.042
	Concave approximation	1647.3086	84	17.422
	Outer approximation CC	1710.0609	120	≥ 10,000
	Outer approximation CP	2581.0224	2	≥ 10,000
	Outer approximation dual	1663.0632	53	≥ 10,000
	MILO	1607.3254	52	≥ 10,000
	MILO-BCD	**1589.5884**	37	8553.430
w2a	Forward stepwise	395.0422	51	283.327
	Backward stepwise	479.2162	169	137.361
	LASSO	721.0487	294	≥ 10,000
	Penalty decomposition	1973.6854	1	≥ 10,000
	Concave approximation	534.2647	166	144.470
	Outer approximation CC	760.3417	301	≥ 10,000
	Outer approximation CP	833.2059	7	≥ 10,000
	Outer approximation dual	722.9003	294	207.279
	MILO	358.5662	82	≥ 10,000
	MILO-BCD	**339.7765**	55	≥ 10,000

Table 3 (continued)

Dataset	Method	AIC	ℓ_0	Time (s)
madelon	Forward stepwise	2506.9165	91	461.957
	Backward stepwise	2528.5802	156	431.609
	LASSO	2523.0742	103	1795.424
	Penalty decomposition	2638.5021	4	833.624
	Concave approximation	2769.9642	340	47.042
	Outer approximation CC	2652.4555	9	≥ 10,000
	Outer approximation CP	2765.2852	2	≥ 10,000
	Outer approximation dual	2657.4810	15	≥ 10,000
	MILO	2616.5531	16	≥ 10,000
	MILO-BCD	**2504.0655**	102	≥ 10,000

sets are employed. We can see that in some cases a slightly larger value of b allows to retrieve even better solutions than those obtained in the experiments of Sect. 4, but the computational cost significantly increases. In the end, as can be observed in Sect. 4, the choice $b = 20$ experimentally led to excellent results on the entirety of our benchmark.

5 Conclusions

In this paper, we considered the problem of best subset selection in logistic regression, with particular emphasis on the IC-based formulation. We introduced an algorithm combining mixed-integer programming models and decomposition techniques like the block coordinate descent. The aim of the algorithm is to find high quality solutions even on larger scale problems, where other existing MIP-based methods are unreasonably expensive, while heuristic and local-optimization-based methods produce very poor solutions.

We theoretically characterized the features and the behavior of the proposed method. Then, we showed the results of wide computational experiments, proving that the proposed approach indeed is able to find, in a reasonable running time, much better solutions than a set of other state-of-the-art solvers; this fact appears particularly evident on the problems with higher dimensions.

Future research will be focused on the definition of possibly more effective and efficient working set selection rules for our algorithm. Upcoming work may also be aimed at adapting the proposed algorithm to deal with different or more general problems.

In particular, the case of multi-class classification is of great interest. However, the problem is challenging. Specifically, the complexity in directly extending our approach to the multinomial case lies in the definition of the piece-wise linear approximation of the objective function. Indeed, in the multi-class scenario,

Table 4 Results of BIC minimization in logistic regression with different optimization methods on small datasets (best result for each dataset in bold)

Dataset	Method	BIC	ℓ_0	Time (s)
parkinsons	Forward stepwise	142.4486	3	0.198
	Backward stepwise	166.7417	12	0.165
	LASSO	140.6959	2	6.391
	Penalty decomposition	277.1788	1	25.056
	Concave approximation	147.2337	4	2.922
	Outer approximation CC	139.1962	6	$\geq$ 10,000
	Outer approximation CP	139.1962	6	5533.104
	Outer approximation dual	145.6313	3	3.412
	MILO	137.6446	6	16.276
	MILO-BCD	**137.6011**	6	93.572
heart	Forward stepwise	225.7059	5	0.337
	Backward stepwise	279.0788	19	0.065
	LASSO	227.2449	5	2.350
	Penalty decomposition	317.3171	1	61.155
	Concave approximation	251.8435	12	2.156
	Outer approximation CC	236.8324	3	$\geq$ 10,000
	Outer approximation CP	288.0285	5	$\geq$ 10,000
	Outer approximation dual	234.3866	7	4.045
	MILO	223.7984	6	22.865
	MILO-BCD	**223.6797**	6	116.618
breast	Forward stepwise	195.8299	2	0.216
	Backward stepwise	265.9623	32	0.354
	LASSO	195.8299	2	21.225
	Penalty decomposition	195.8299	2	7.972
	Concave approximation	203.5140	6	3.201
	Outer approximation CC	194.2193	4	$\geq$ 10,000
	Outer approximation CP	231.7141	8	$\geq$ 10,000
	Outer approximation dual	195.8300	2	6.664
	MILO	**192.4211**	10	653.077
	MILO-BCD	193.5695	5	124.760
biodeg	Forward stepwise	782.2522	13	2.560
	Backward stepwise	792.7384	26	0.400
	LASSO	785.1660	22	65.498
	Penalty decomposition	950.2008	4	420.761
	Concave approximation	772.6349	22	6.917
	Outer approximation CC	880.5540	12	$\geq$ 10,000
	Outer approximation CP	1154.3736	5	$\geq$ 10,000
	Outer approximation dual	835.4689	31	29.016
	MILO	746.8531	14	$\geq$ 10,000
	MILO-BCD	**745.1778**	13	2305.093

Table 4 (continued)

Dataset	Method	BIC	ℓ_0	Time (s)
spectf	Forward stepwise	203.9442	4	0.573
	Backward stepwise	237.7547	17	0.252
	LASSO	208.8296	7	8.949
	Penalty decomposition	277.1788	1	25.056
	Concave approximation	228.256224	12	3.867
	Outer approximation CC	214.4389	3	≥ 10,000
	Outer approximation CP	224.2627	5	≥ 10,000
	Outer approximation dual	251.6325	7	10.649
	MILO	196.8356	5	231.938
	MILO-BCD	**196.8238**	5	115.597
libras	Forward stepwise	101.5028	4	1.145
	Backward stepwise	270.8327	46	0.970
	LASSO	71.4674	9	5.453
	Penalty decomposition	182.2357	1	42.429
	Concave approximation	131.7340	17	14.591
	Outer approximation CC	75.3065	9	≥ 10,000
	Outer approximation CP	153.6910	5	≥ 10,000
	Outer approximation dual	132.1737	10	58.047
	MILO	**41.3979**	7	≥ 10,000
	MILO-BCD	53.0895	9	642.618

the number of weights is $n \times m$, being m the number of classes, and $N \times m$ pieces of the objective function need to be approximated. This results in an increasingly high number of variables and constraints to be handled, which might become rapidly unmanageable even exploiting our decomposition approach. Hence, future work might be focused on devising alternative decomposition approaches specifically designed to tackle the multinomial case.

Appendix: Review of related algorithms

A number of techniques has been proposed and considered in the literature to tackle problem (3). If the number of variables n in not exceedingly large, especially in the case of convex $\mathcal{L}$, heuristic and even exhaustive approaches are a viable way of proceeding.

The *exhaustive approach* consists of finding the global minimum for $\mathcal{L}$ for all possible combinations of non-zero variables. All the retrieved solutions are then compared, adding to $\mathcal{L}$ the penalty term on the ℓ_0-norm, to identify the optimal

Table 5 Results of BIC minimization in logistic regression with different optimization methods on large datasets (best result for each dataset in bold)

Dataset	Method	BIC	ℓ_0	Time (s)
spam	Forward stepwise	2361.1337	27	28.446
	Backward stepwise	2140.7302	32	2.0124
	LASSO	2177.5219	40	1214.969
	Penalty decomposition	6184.8715	1	≥ 10,000
	Concave approximation	2196.5090	38	16.464
	Outer approximation CC	2336.2203	58	≥ 10,000
	Outer approximation CP	3894.7351	10	≥ 10,000
	Outer approximation dual	2275.2687	51	157.805
	MILO	2150.2450	30	≥ 10,000
	MILO-BCD	**2137.9834**	31	≥ 10,000
digits	Forward stepwise	552.1658	13	8.110
	Backward stepwise	468.9395	20	2.615
	LASSO	529.0165	24	2160.987
	Penalty decomposition	5299.8033	0	≥ 10,000
	Concave approximation	516.442699	28	32.288
	Outer approximation CC	640.2697	10	≥ 10,000
	Outer approximation CP	1696.2871	5	≥ 10,000
	Outer approximation dual	596.1621	28	128.011
	MILO	448.3050	14	≥ 10,000
	MILO-BCD	**441.0145**	15	9949.433
a2a	Forward stepwise	1741.3958	15	10.727
	Backward stepwise	2016.2528	64	5.798
	LASSO	1764.5871	15	397.503
	Penalty decomposition	1860.2444	5	607.869
	Concave approximation	1873.3706	44	21.709
	Outer approximation CC	2028.2982	11	≥ 10,000
	Outer approximation CP	2268.7472	4	≥ 10,,000
	Outer approximation dual	1829.5696	14	≥ 10,000
	MILO	1754.9999	16	≥ 10,000
	MILO-BCD	**1733.8513**	17	2933.3452
w2a	Forward stepwise	614.3182	18	107.705
	Backward stepwise	1320.8765	143	147.459
	LASSO	2524.0133	293	≥ 10,000
	Penalty decomposition	1979.8373	1	≥ 10,000
	Concave approximation	919.0693	70	166.238
	Outer approximation CC	879.0359	2	≥ 10,000
	Outer approximation CP	931.5931	3	≥ 10,000
	Outer approximation dual	2531.5618	294	205.223
	MILO	671.9868	20	≥ 10,000
	MILO-BCD	**579.0229**	26	8842.6791

Table 5 (continued)

Dataset	Method	BIC	ℓ_0	Time (s)
`madelon`	Forward stepwise	**2660.6283**	3	24.179
	Backward stepwise	2732.3224	15	488.801
	LASSO	2661.9344	6	1852.270
	Penalty decomposition	2772.5887	0	75.713
	Concave approximation	3030.0118	86	152.799
	Outer approximation CC	2677.8156	4	≥ 10,000
	Outer approximation CP	2781.6611	2	≥ 10,000
	Outer approximation dual	2689.3907	2	≥ 10,000
	MILO	2681.9310	1	≥ 10000
	MILO-BCD	**2660.6283**	3	≥ 10,000

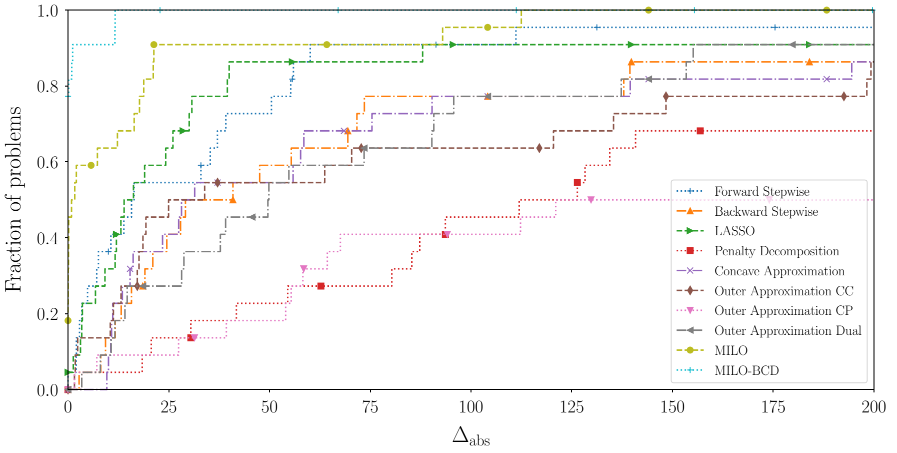


Fig. 1 Each curve represents the fraction of the 22 classification problems for which the corresponding solver obtains an absolute error less or equal than Δ_{abs} w.r.t. the optimal value

solution to the original problem. This approach is however clearly computationally intractable.

In applications, an heuristic relaxation of the exhaustive search is employed: the greedy *step-wise approach*, with both its variants, the *forward selection* strategy and the *backward elimination* strategy [17]. This method consists of adding (or removing, respectively) a variable to the support, in such a way that the variation of the objective function obtained by only changing that variable is optimal; the procedure typically stops as soon as the addition (removal) of a variable is not enough to improve the quality of the solution. This technique is clearly much cheaper, at the cost of a lower quality of the final solution retrieved.

Table 6 Results obtained by the MILO-BCD procedure on the best subset selection problem based on AIC with four datasets for different values of working set size b

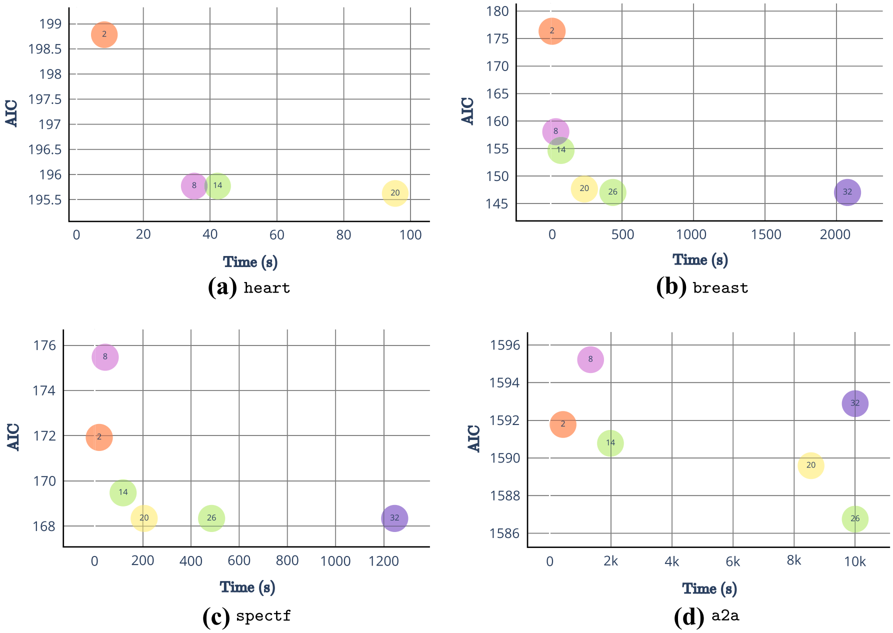
Dataset	b	AIC	ℓ_0	Time (s)
`heart`	2	198.7826	11	8.326
	8	195.7715	10	35.240
	14	195.7715	10	42.222
	20	195.6242	11	95.399
	26	–	–	–
	32	–	–	–
`breast`	2	176.3391	7	10.079
	8	158.0725	13	36.012
	14	154.6846	17	72.4643
	20	147.6781	17	236.126
	26	147.0381	19	435.3751
	32	147.0381	19	2077.4473
`spectf`	2	171.9253	12	19.333
	8	175.4713	7	43.999
	14	169.4771	18	118.313
	20	168.3443	15	205.6255
	26	168.3443	15	485.486
	32	168.3443	15	1245.422
`a2a`	2	1591.7767	35	430.714
	8	1595.2172	37	1333.7368
	14	1590.7749	35	1984.113
	20	1589.5884	37	8553.430
	26	1586.7499	39	≥ 10,000
	32	1592.8824	37	≥ 10,000

One of the most prominent approaches (arguably the most popular one) to induce sparsity in model estimation problems is Lasso [41]. Lasso consists of approximating the ℓ_0 penalty term by a continuous, convex surrogate, the ℓ_1-norm. When applied to (3), the resulting optimization problem is the widely used ℓ_1-regularized formulation of logistic regression [27, 29, 45]:

$$\min_{w\in\mathbb{R}^n} \mathcal{L}(w) + \lambda\|w\|_1. \tag{14}$$

The ℓ_1-norm is well known to be sparsity-inducing [3]. Lasso often produces good solutions with a reasonable computational effort and is particularly suited for large scale problems, where methods directly tackling the ℓ_0 formulation are too expensive to be employed. However, equivalence relationships between problems (3) and (14) do not exist. Thus, problem (14) usually has to be solved for many different values of λ in order to find a satisfying solution of (14). Still, the solution is typically suboptimal for problem (3) and poor from the statistical point of view [33, 40, 46].

Lu and Zhang [32] proposed a *Penalty Decomposition* (PD) approach to solve problem (3). The classical variable splitting technique [25] can be applied to problem (3), duplicating the variables, adding a linear equality constraint



(a) `heart` **(b)** `breast`

(c) `spectf` **(d)** `a2a`

Fig. 2 Trade-off between runtime and solution quality for different values of the working set size in MILO-BCD, on the best subset selection problem based on AIC for the four considered problems

and separating the two parts of the objective function, obtaining the following problem:

$$\begin{aligned} \min_{w,z\in\mathbb{R}^n} \; & \mathcal{L}(w) + \lambda\|z\|_0 \\ \text{s.t. } & w - z = 0. \end{aligned} \tag{15}$$

Problem (15) can then be solved by an alternate exact minimization of the quadratic penalty function

$$q_\tau(w,z) = \mathcal{L}(w) + \lambda\|z\|_0 + \frac{\tau}{2}\|w-z\|_2^2, \tag{16}$$

where the penalty parameter τ is increased every time a (approximate) stationary point, w.r.t. the w block of variables, of the current q_τ is attained. The algorithm is summarized in Algorithm 2.

Algorithm 2: Penalty Decomposition

```
Input: $\tau > 0,\ \sigma_\tau > 1,\ w^0, z^0 \in \mathbb{R}^n,\ \varepsilon > 0,\ \eta > 0,\ \sigma_\varepsilon \in (0,1)$.
$k = 0$
while $\|w^k - z^k\| > \eta$ do
    Set
        $w^{k+1} = \arg\min_w \mathcal{L}(w) + \frac{\tau}{2}||w - z^k||^2$
    Set
        $z^{k+1} = \arg\min_z \frac{\tau}{2}||w^{k+1} - z||^2 + \lambda||z||_0$
    if $\|\nabla_w q_\tau(w^k, z^k)\| \leq \varepsilon$ then
        Set $\tau = \sigma_\tau \tau$
        Set $\varepsilon = \sigma_\varepsilon \varepsilon$
    $k = k + 1$
return $z^k$
```

The z-update step can in fact be carried out in closed form by the following rule:

$$z_i^{k+1} = \begin{cases} 0 & \text{if } \frac{\tau}{2}(w_i^k)^2 < \lambda, \\ w_i^{k+1} & \text{otherwise.} \end{cases}$$

The algorithm is proved to asymptotically converge to Lu–Zhang stationary points, i.e., to local minima. The solution retrieved by the algorithm strongly depends on the choice of the initial value of the penalty parameter τ and of the increase factor σ_τ. Therefore, in order to find good quality solutions, the algorithm may be run in practice several times with different hyperparameters configurations.

A different approach exploits the fact that the ℓ_0 semi-norm can be approximated by the sum of a finite sum of scalar terms, each one being a surrogate for the step function. In particular, the scalar step function can be approximated, for $t > 0$, by the continuously differentiable concave function $s(t) = 1 - e^{-\alpha t}$, as done by [37] or [31]. Problem (3) can hence be reformulated as

$$\min_{w \in \mathbb{R}^n} \mathcal{L}(w) + \lambda \sum_{i=1}^{n} (1 - e^{-\alpha |w_i|}). \tag{17}$$

A sequence of problems of the form (17), for increasing values of α, can then be solved, producing a sequence of solutions that are increasingly good approximations of those of the original problem. In fact, in the computational practice, problem (17) is solved for a suitable, fixed value of α.

In recent years, very effective algorithms have been proposed in the literature to tackle the sparse logistic regression in its cardinality-constrained formulation, i.e., to solve the problem

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \ & \mathcal{L}(w) \\ \text{s.t.} \ & \|w\|_0 \leq s, \end{aligned} \tag{18}$$

for fixed $s < n$. Among these methods, the most remarkable one is arguably the *Outer Approximation method* [10, 16], which was proposed to be used for problem (18) by [8]. The algorithm, which is briefly reported in Algorithm 3, works in an alternating minimization fashion. First, it exactly solves, through a mixed-integer solver, a cutting-plane based approximation of the problem; then, it finds the exact global minimum w.r.t. the support of the newly obtained solution. If the objective function of the MIP problem is within some pre-specified tolerance ϵ of the true objective function at the new iterate, then the algorithm stops, otherwise the obtained point is used to perform a new cut.

Algorithm 3: Outer Approximation Method

```
Input: $M \gg 0$, $w^0 \in \mathbb{R}^n$, $\nu^0 = -\infty$, $\epsilon > 0$.
$k = 0$
while $\nu^k - \mathcal{L}(w^k) < \epsilon$ do
    Set
        $\hat{\beta}, \nu^{k+1} \in \arg\min_{\beta, w} \beta$
            s.t. $-Mz_i \leq w_i \leq Mz_i \quad \forall\, i = 1, \dots, n,$
                 $z \in \{0, 1\}^n,$
                 $\sum_{i=1}^{n} z_i \leq s,$
                 $\beta \geq \mathcal{L}(w^\ell) + \nabla \mathcal{L}(w^\ell)^T (w - w^\ell) \quad \forall\, \ell = 0, \dots, k,$
    Set
        $w^{\ell+1} \in \arg\min_{w} \mathcal{L}(w)$
            s.t. $w_i = 0$ for all $i \in \bar{S}(\nu^{k+1})$
    $k = k + 1$
return $z^k$
```

Algorithm 3 can be employed to solve problem (3), by running it for every possible value of $s = 1, \dots, n$ and choosing, among the n retrieved solutions, the one with lowest IC value.

In fact, the algorithm can straightforwardly be adapted to directly handle problem (3). To this aim it is sufficient to remove from the MIP subproblem the cardinality constraint and add it as a penalty term in the objective function.

Recently, Kamiya et al. [26] proposed an alternative way of using the outer approximation method, which is however based on the ℓ_2-regularized formulation of the logistic regression problem with cardinality constraints

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \; & \mathcal{L}(w) + \frac{1}{2\gamma} \|w\|_2^2 \\ \text{s.t.} \; & \|w\|_0 \leq s. \end{aligned}$$

Applying duality theory, the optimal value obtainable for a fixed configuration z of nonzero variables, $c(z)$, can be computed by solving the problem

$$\begin{aligned}\max_{\alpha\in\mathbb{R}^N} \; & -\sum_{i=1}^{N}\big(\alpha_i\log(\alpha_i)+(1-\alpha_i)\log(1-\alpha_i)\big)-\frac{\gamma}{2}\sum_{j=1}^{n} z_j\left(\sum_{i=1}^{N} y_i\alpha_i X_{ij}\right)^2\\ \text{s.t. } & \alpha\in[0,1]^N\end{aligned}$$

whereas cuts for the cutting-planes approximation can be added as

$$\beta \geq c(z^{\ell}) + \nabla c(z^{\ell})^T(z-z^{\ell}),$$

where

$$\frac{\partial c(z)}{z_j} = -\frac{\gamma}{2} z_j\left(\sum_{i=1}^{N} y_i\alpha_i X_{ij}\right)^2.$$

They also show that the left hand side of the objective function in the dual problem can be approximated by a properly defined parabola, which makes the problem quadratic and thus much more efficiently solvable:

$$\alpha\log(\alpha)+(1-\alpha)\log(1-\alpha) \approx \frac{5}{2}\alpha^2-\frac{5}{2}\alpha-\frac{1}{12}$$

This approximation, seen back in the primal space, is a good quadratic piecewise approximation of the logistic loss which should be more accurate than the piecewise linear employed by [38].

Acknowledgements The authors would like to thank L. Di Gangi for the useful discussions. We would also like to thank the associate editor and the anonymous referees whose constructive suggestions led us to improve the quality of this manuscript. Moreover, our appreciation goes to Dr. S. Kamiya for kindly providing us the code used in his work [26]. Finally, we thank Gurobi for the academic licence to our Global Optimization Laboratory.

Funding Open access funding provided by Università degli Studi di Firenze within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**(6), 716–723 (1974)
2. Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Selected Papers of Hirotugu Akaike, pp. 199–213. Springer (1998)
3. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Found. Trends Mach. Learn. **4**(1), 1–106 (2012). https://doi.org/10.1561/2200000015
4. Beck, A., Eldar, Y.: Sparsity constrained nonlinear optimization: optimality conditions and algorithms. SIAM J. Optim. **23**(3), 1480–1509 (2013)
5. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and distributed computation: numerical methods, vol. 23. Prentice Hall, Englewood Cliffs (1989)
6. Bertsimas, D., Digalakis Jr, V.: The backbone method for ultra-high dimensional sparse machine learning. arXiv:2006.06592 (2020)
7. Bertsimas, D., King, A., Mazumder, R.: Best subset selection via a modern optimization lens. Ann. Stat. **44**, 813–852 (2016)
8. Bertsimas, D., King, A., et al.: Logistic regression: from art to science. Stat. Sci. **32**(3), 367–384 (2017)
9. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
10. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., et al.: An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optim. **5**(2), 186–204 (2008)
11. Bozdogan, H.: Akaike's information criterion and recent developments in information complexity. J. Math. Psychol. **44**(1), 62–91 (2000)
12. Burnham, K.P., Anderson, D.R.: Practical use of the information-theoretic approach. In: Model Selection and Inference, pp. 75–117. Springer (1998)
13. Burnham, K.P., Anderson, D.R.: Multimodel inference: understanding AIC and BIC in model selection. Sociol. Methods Res. **33**(2), 261–304 (2004)
14. Di Gangi, L., Lapucci, M., Schoen, F., Sortino, A.: An efficient optimization approach for best subset selection in linear regression, with application to model selection and fitting in autoregressive time-series. Comput. Optim. Appl. **74**(3), 919–948 (2019)
15. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
16. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986)
17. Efroymson, M.A.: Multiple regression analysis. In: Ralston, A., Wilf, H.S. (eds.) Mathematical Methods for Digital Computers, pp. 191–203. Wiley, New York (1960)
18. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
19. Gómez, A., Prokopyev, O.A.: A mixed-integer fractional optimization approach to best subset selection. INFORMS J. Comput. (2021, accepted)
20. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020). http://www.gurobi.com
21. Hannan, E.J., Quinn, B.G.: The determination of the order of an autoregression. J. R. Stat. Soc. Ser. B (Methodol.) **41**(2), 190–195 (1979)
22. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Berlin (2009)
23. Hosmer, D.W., Jr., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression, vol. 398. Wiley, Hoboken (2013)
24. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning, vol. 112. Springer, Berlin (2013)
25. Jörnsten, K., Näsberg, M., Smeds, P.: Variable Splitting: A New Lagrangean Relaxation Approach to Some Mathematical Programming Models. LiTH MAT R.: Matematiska Institutionen. University of Linköping, Department of Mathematics (1985)
26. Kamiya, S., Miyashiro, R., Takano, Y.: Feature subset selection for the multinomial logit model via mixed-integer optimization. In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 1254–1263 (2019)
27. Kim, S.-J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale ℓ_1-regularized least squares. IEEE J. Sel. Top. Signal Process. **1**(4), 606–617 (2007)

28. Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling. Springer, Berlin (2008)
29. Lee, S.-I., Lee, H., Abbeel, P., Ng, A.Y.: Efficient ℓ_1 regularized logistic regression. AAAI **6**, 401–408 (2006)
30. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1–3), 503–528 (1989)
31. Liuzzi, G., Rinaldi, F.: Solving ℓ_0-penalized problems with simple constraints via the Frank–Wolfe reduced dimension method. Optim. Lett. **9**(1), 57–74 (2015)
32. Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. SIAM J. Optim. **23**(4), 2448–2478 (2013)
33. Miller, A.: Subset Selection in Regression. Chapman and Hall/CRC, Boca Raton (2002)
34. Miyashiro, R., Takano, Y.: Mixed integer second-order cone programming formulations for variable selection in linear regression. Eur. J. Oper. Res. **247**(3), 721–731 (2015a)
35. Miyashiro, R., Takano, Y.: Subset selection by Mallows' Cp: a mixed integer programming approach. Expert Syst. Appl. **42**(1), 325–331 (2015b)
36. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
37. Rinaldi, F., Schoen, F., Sciandrone, M.: Concave programming for minimizing the zero-norm over polyhedral sets. Comput. Optim. Appl. **46**(3), 467–486 (2010)
38. Sato, T., Takano, Y., Miyashiro, R., Yoshise, A.: Feature subset selection for logistic regression via mixed integer optimization. Comput. Optim. Appl. **64**(3), 865–880 (2016)
39. Schwarz, G., et al.: Estimating the dimension of a model. Ann. Stat. **6**(2), 461–464 (1978)
40. Shen, X., Pan, W., Zhu, Y., Zhou, H.: On constrained and regularized high-dimensional regression. Ann. Inst. Stat. Math. **65**(5), 807–832 (2013)
41. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**(1), 267–288 (1996)
42. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory Appl. **109**(3), 475–494 (2001)
43. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K. Jarrod, Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E.W., Vander Plas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P.: Fundamental algorithms for scientific computing in python and SciPy 1.0 contributors. SciPy 1.0. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
44. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero-norm with linear models and kernel methods. J. Mach. Learn. Res. **3**(Mar), 1439–1461 (2003)
45. Yuan, G.-X., Ho, C.-H., Lin, C.-J.: An improved GLMNET for L1-regularized logistic regression. J. Mach. Learn. Res. **13**(64), 1999–2030 (2012)
46. Zheng, Z., Fan, Y., Lv, J.: High dimensional thresholded regression and shrinkage effect. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **76**(3), 627–649 (2014)