An Efficient Optimization Approach for Best Subset Selection in Linear Regression, with Application to Model Selection and Fitting in

Noname manuscript No.
(will be inserted by the editor)

An Efficient Optimization Approach for Best Subset Selection in Linear Regression, with Application to Model Selection and Fitting in Autoregressive Time-Series

L. Di Gangi[a] · M. Lapucci[a] · F. Schoen[a] · A. Sortino[a]

the date of receipt and acceptance should be inserted later

Abstract In this paper we consider two relevant optimization problems: the problem of selecting the best sparse linear regression model and the problem of optimally identifying the parameters of auto-regressive (AR) models based on time series data. Usually these problems, which although different are indeed related, are solved through a sequence of separate steps, alternating between choosing a subset of features and then finding a best fit regression.

In this paper we propose to model both problems as mixed integer non linear optimization ones and propose numerical procedures based on state of the art optimization tools in order to solve both of them. The proposed approach has the advantage of considering both model selection as well as parameter estimation as a single optimization problem. Numerical experiments performed on widely available datasets as well as on synthetic ones confirm the high quality of our approach, both in terms of the quality of the resulting models and in terms of CPU time.

Keywords:

MINLP, Linear Regression, Subset Selection, Autoregressive Time-series

1 Introduction

In this paper we consider two estimation problems which, although quite similar, possess some characteristic features which require some specific care.

The first problem we consider is that of selecting the best subset of features in a linear regression problem. By this we mean that, differently from standard regression, where a best linear model is looked for regressing a real observed variable against a finite set of independent features, here the modeler would like to fit a good linear model by using a relatively small number of features. Of course a compromise needs to be accepted between a very sparse model and a

✉ F. Schoen

E-mail: fabio.schoen@unifi.it

[a] Department of Information Engineering, Università degli Studi di Firenze, Via di Santa Marta 3, 50139, Firenze (Italy)

very high quality of fit. In statistics many quality of fit measures have been traditionally defined in order to be able to compare models depending on different subsets of features.

The second problem we will consider is that of fitting a low-order auto-regressive model to a dataset generated by a time series. Here, again, the problem is to find a best regression model of the value of the series at time t as a function of the same series observed at time instants preceding t. In auto-regressive (AR) models the regression considers all previous values of the observed variable up to a distance p in the past, which is called the order of the AR model. In AR time series model fitting, it is required to estimate the order of the model, together with the coefficients of the regression.

Both the above problems share a fitting phase, which is usually quite easy to accomplish by means of convex optimization solvers, and a model choice phase which, in the first problem, consists of identifying a "best" subset of features, while in the second one consists in finding the best order of the AR model. This second phase has a combinatorial nature and usually greatly increases the difficulty of the overall model selection and estimation problem. In fact, the usual approach in tackling these problems is to decompose the problem into two phases and, usually, calling an heuristic procedure to deal with the combinatorial component.

In this paper we propose to exploit the increasing availability and quality of non linear integer programming solvers in order to tackle the whole estimation and selection phases as a single optimization problem.

Best subset selection in linear regression

The problem of best subset selection in regression is a fundamental problem in statistics, where it is frequently defined as the variable selection problem, as well as in machine learning, where the term feature selection is more frequently adopted. In standard regression a set of "dependent variables" and covariates are observed and best fit is looked for through the solution of an optimization problem like, e.g., a convex least squares one. The best subset selection variant of the problem requires also to select a small subset of variables (or features) among the available ones in order to obtain a sparse regression model. The advantages of sparsity are many: a model depending on a small set of variables might offer a better explanation of the underlying generating model; moreover, sparsity reduces the risk of overfitting. For a nice introduction to this topic we refer the reader to [29]. The problem can be defined as follows: a set of N observations of P-variate data $\{x_{ij}\}$ is available as well as a vector $Y \in \mathbb{R}^N$ of response variables or co-variates. The linear regression problem consists in finding coefficients β, c so that

$$y_i \approx \sum_{j=1}^{P} \beta_j x_{ij} + c \qquad\qquad i = 1, \ldots, N.$$

Choosing the best subset requires two important steps: first a quality of fit measure needs to be defined, which takes into account both the approximation error as well as the model complexity (associated to the cardinality of the set of non null coefficients). Second, as the problem has both a continuous and a discrete nature, a suitable optimization algorithm needs to be defined. While from the modeling point of view several measures have been proposed in the literature which are nowadays commonly accepted in the scientific community, much less effort has been devoted to the algorithmic

aspects. Most approaches are based on a strict separation of the estimation problem, which corresponds to an easy convex optimization problem, from the combinatorial aspect connected to the suitable choice of a subset of significant features. This separation, of course, contributes to lowering the complexity of the overall algorithm at the expense of losing optimality. Most classical approaches are based either on an expensive exhaustive search, feasible only when the number of features is small, or on greedy approaches, like those based on step-wise regression, based on sequential addition or deletion of single feature variables from the set of candidates. Other approaches are based on avoiding cardinality constraints by adding convex penalties like, e.g., the L^1 norm of the coefficients to be estimated, in an attempt to favor the sparsity, following the well known Lasso approach [38]. Recently, thanks also to the availability of powerful mixed integer nonlinear (quadratic or conic, in particular) solvers, some authors (see, e.g., [7, 29, 30, 19]) have started proposing approaches based on the direct solution of the whole problem; a similar approach, for a slightly different regression model, was also proposed in [33]. These alternative approaches are very interesting, as they try to solve the whole problem as a single one and are based either on the sequential solution of different cardinality constrained regression problems [7], or on the reformulation of the whole problem as a single mixed integer second order conic optimization model [30] or as a mixed integer fractional optimization one [19].

Model selection and fitting in AR time series

A time series $\{X_t\}$ is generated by a stationary AR(p) model if the following linear relationship holds:

$$X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t,$$

where $c \in \mathbb{R}$ is the intercept term and $(\varphi_1, \ldots, \varphi_p) \in \mathbb{R}^p$ are autoregressive coefficients which explain the temporal dynamics of the series; these coefficients are required to satisfy some stationarity constraints. Finally, ε_t are noise terms.

The selection of the AR order is a non trivial task. The Box and Jenkins procedure suggests the identification of the order p through the analysis of the estimated autocorrelation function (ACF) and the partial autocorrelation function (PACF). This procedure however is highly subjective and requires statistical knowledge. Moreover, it is not feasible when several AR models have to be fitted automatically, in an on line, real time system. In this scenario, automatic algorithms that perform model selection and fitting of autoregressive models are required. The `R` function `auto.arima()` implements one of these algorithms and conducts a search over possible autoregressive models within the provided order constraints.

Model selection is performed usually based on a given information criterion (AIC [1], HQIC [22], BIC [34]). More details can be found in [23].

Optimization approaches to solve classical statistical problems have been proposed in recent literature – see, e.g., [27, 4, 6, 7]. The main idea in these approaches is that thanks to the solution of a suitable optimization model, the identification of the model and the parameter estimation can be performed through a single global optimization algorithm, thus avoiding the sub-optimality of classical approaches, which usually proceed in sequential stages. While it is well known that model identification and parameter estimation are indeed optimization problems, traditionally the statistical community has

not yet fully exploited the possibility of modeling the whole process as a single optimization problem, mainly due to the inherent difficulties of non convex optimization. However, the power of modern optimization approaches enables us to experiment with global optimization techniques in this framework. In this paper, a novel optimization approach to automatically perform model selection and fitting is proposed for the autoregressive case. The reasons why we are proposing an optimization approach for autoregressive models only (and not, e.g., the more general ARMA models) will become clear later. We can anticipate that this restriction, which we plan to remove in a future paper, gives us the possibility of proving the efficiency of our proposal, without introducing too many technical details.

Outline of the contributions of this paper

In this paper, for both problems, we follow a similar approach, based on the consideration that mixed integer quadratic optimization, although hard in general, can be quite safely exploited when the dimension of the problem is "sufficiently small". The basic idea of this paper has been that of using a decomposition approach in which variables are divided into two blocks, one of which reduces to the solution of a quadratic integer problem. The approach proposed in this paper has been analyzed from a theoretical point of view, and convergence guarantees have been obtained. Moreover, when tested on widely used test sets, our algorithm displayed excellent performance with respect to state of the art methods, both for the subset selection as well as for the AR time series modeling. The proposed approach is fast and reliable and, although not guaranteed to deliver the globally optimal solutions (mainly when the underlying integer solver is artificially stopped before convergence), it generates very good solutions in very low computational time.

Moreover, in AR time series model estimation, the approach we propose is capable of dealing, at least partially, with the stationarity problem.

The structure of the paper is the following. In section 2 we formally define the best subset linear regression problem, introduce different quality of fit criteria and recall some of the models proposed for this problem. In section 3 we introduce our proposal for a method designed to solve the subset selection problem. A convergence proof is also given. In section 4 we define a model for the model selection and estimation problem in auto-regressive time series and we show how the model can be formulated in a way that fits the capabilities of the algorithm introduced in the previous section. Finally, section 5 reports results from an extensive computational comparison of our proposed algorithm against the best performing methods in the literature for both problems.

2 Best Subset Selection in Linear Regression

Given a dataset $X = [x_1, \ldots, x_P] \in \mathbb{R}^{N \times P}$ composed of N samples of P features and a vector $Y \in \mathbb{R}^N$ of response variables, the linear regression setting models the relationship between them as

$$y_i = \sum_{j=1}^{P} \beta_j x_{ij} + c + \epsilon_i \qquad i = 1, \ldots, N \tag{1}$$

where $\beta \in \mathbb{R}^P$ and $c \in \mathbb{R}$ are the model parameters and $\epsilon \in \mathbb{R}^N$ is a vector of independent random errors, whose expected values are assumed to be zero and their variances σ^2 are constant among the observations. Frequently in applications the modeler is required to solve a sparse version of the above problem, that is, to find a subset of $k < P$ features which gives the best fit to the response y according, e.g., to the squared error measure. This problem is known as the *best subset selection* and can be formalized as follows:

$$\begin{aligned} \min_{c\in\mathbb{R},\ \beta\in\mathbb{R}^P} \ & \sum_{i=1}^{N} \left(y_i - c - \sum_{j=1}^{P} \beta_j x_{ij} \right)^2 \\ & \|\beta\|_0 \leq k, \end{aligned} \tag{2}$$

where $\|v\|_0$ is the zero semi-norm of v (i.e., the number of non zero elements in v) and the sparsity parameter k is a priori fixed. Since this problem is known to be NP-hard [31], exactly solving it might require a large computational effort, comparable to that of an exhaustive, brute-force approach. In order to overcome such inefficiency, approximation approaches like Lasso [38] and forward step-wise selection [17] have been proposed in literature. However, step-wise selection is an heuristic which can often lead to suboptimal choices of variables. On the other hand, Lasso optimizes a surrogate model (it replaces the 0-seminorm with the convex 1-norm): this could lead to solutions which are not sufficiently sparse or to solutions which are poor from the statistical point of view [28,35,40].

More recently, approaches based on Mixed Integer Optimization (MIO) have been introduced to solve (2) [8,29,30,6, 7,9,13,39]. When k is given, introducing binary indicator variables $z \in \{0,1\}^P$ s.t $z_i = 0$ only if $\beta_i = 0$, problem (2) can be reformulated as:

$$\begin{aligned} \min_{c\in\mathbb{R},\ \beta\in\mathbb{R}^P} \ & \sum_{i=1}^{N} \left(y_i - c - \sum_{j=1}^{P} \beta_j x_{ij} \right)^2, \\ & e^T z \leq k, \\ & -Mz \leq \beta \leq Mz, \\ & z \in \{0,1\}^P, \beta \in \mathbb{R}^P, \end{aligned} \tag{3}$$

where e is the vector of P elements all equal to 1 and M is a sufficiently large constant.

When the parameter k is not fixed, i.e. it is an additional variable of the problem, as in typical applications, the best subset selection problem becomes more challenging. First, a Goodness-Of-Fit (GOF) measure has to be introduced in evaluating regression models based on subset of variables with different cardinalities. In this regard, we need to assume normal independent errors ϵ_i. The problem of choosing a reasonable compromise between sparsity and accuracy is clearly a bi–objective one; in the literature this has been usually approached by means of a surrogate single objective function. Among the commonly used formulations we recall:

– **AIC:** The Akaike Information Criterion [2,11], defined as

$$-2\ell(\beta, c, \sigma^2) + 2(\|\beta\|_0 + 2),$$

where $\ell(\beta, c, \sigma^2)$ is the log-likelihood of the model. The AIC provides an estimate of the expected, relative distance between the fitted model and the unknown true mechanism that generated the data. 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 [3], the penalty term $\|\beta\|_0 + 2$, i.e the total number of parameters involved in the model including the intercept term and the variance of the error term, allows to correct this bias.

– **BIC:** The Bayesian Information Criterion, defined as

$$-2\ell(\beta, c, \sigma^2) + \log(N)(\|\beta\|_0 + 2),$$

where $\ell(\beta, c, \sigma^2)$ is the log-likelihood of the model and N is the number of examples used in the fitting process. It has been shown [25,3] 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. The BIC criterion, interpreted as a penalized-likelihood, weighs down model complexity more heavily than the AIC criterion.

– **HQIC:** The Hannan-Quinn Information Criterion [22], defined as

$$-2\ell(\beta, c, \sigma^2) + 2(\|\beta\|_0 + 2)\log(\log N),$$

where $\ell(\beta, c, \sigma^2)$ is the log-likelihood of the model and N is the number of examples used in the fitting process. This criterion provides a consistent estimator of the order of an autoregressive model [25]. HQIC is mainly suited for the autoregression setting.

We recall that the (negative double) log-likelihood of a linear regression model is given by

$$-2\ell(\beta, c, \sigma^2) = N\log(\sigma^2) + N\log(2\pi) + \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(y_i - c - \sum_{j=1}^{P}\beta_j x_{ij}\right)^2.$$

For the above GOF measures, we can use the following general notation:

$$-2\ell(\beta, c, \sigma^2) + \alpha(\|\beta\|_0 + 2),$$

where α depends on the chosen GOF measure. We can therefore consider the following optimization problem:

$$\min_{\beta,\ c,\ \sigma^2} N\log(\sigma^2) + \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(y_i - c - \sum_{j=1}^{P}\beta_j x_{ij}\right)^2 + \alpha(\|\beta\|_0), \tag{4}$$

where we have removed from the objective function the constant terms. By partial differentiation, we can easily derive the maximum likelihood estimator of σ^2, which is

$$\sigma^2 = \frac{R(\beta,\ c)}{N}, \tag{5}$$

being $R(\beta,\ c) = \sum_{i=1}^{N}(y_i - c - \sum_{j=1}^{P}\beta_j x_{ij})^2$. Substituting in (4), we get the following problem:

$$\min_{\beta,\ c}\ N\log\left(\frac{R(\beta, c)}{N}\right) + N + \alpha\|\beta\|_0. \tag{6}$$

Many approaches exist to tackle problem (6). One way, is that of solving $P+1$ problems of the form (2) setting $k = 0, 1, \ldots, P$, either by means of Lasso or by reformulation (3) [13,39,29]. A greedy step-wise procedure, either by forward selection or backward elimination [28], can also be applied, and even exhaustive enumeration [18] is possible on small instances of the problem.

More recently, various approaches have been proposed based on mixed integer models. For instance, a branch-and-bound method has been introduced for AIC minimization [24]. Problem (6) can also be reformulated as the following Mixed-Integer Second Order Cone Programming (MISOCP) problem [29]:

$$\min_{f} \; f \tag{7a}$$

$$\text{s.t. } \varepsilon_i = y_i - \left(c + \sum_{j=1}^{P} \beta_j x_{ij} \right) \quad i = 1, \ldots, N \tag{7b}$$

$$\sum_{i=1}^{N} \varepsilon_i^2 \leq f \cdot \sum_{j=0}^{P} \left(w_j \cdot \exp\left(-\frac{\alpha j}{N} \right) \right) \tag{7c}$$

$$\sum_{j=0}^{P} (j \cdot w_j) = \sum_{j=1}^{P} z_j \tag{7d}$$

$$\sum_{j=0}^{P} w_j = 1 \tag{7e}$$

$$-Mz_j \leq \beta_j \leq Mz_j \quad j = 1, \ldots P \tag{7f}$$

$$\beta \in \mathbb{R}^P,\ c \in \mathbb{R},\ \epsilon \in \mathbb{R}^N,\ f \in \mathbb{R}_+,\ w \in \{0,1\}^{P+1},\ z \in \{0,1\}^P \tag{7g}$$

This mixed-integer model is elegant, but in fact cannot be solved so efficiently. In particular, this method is in practice much slower not only w.r.t. the step-wise heuristic, but also with respect to the approach solving $P+1$ times problem (3) [29,19]. The latter method is reported as the most efficient method to date [19].

Recently, a Mixed-Integer Fractional formulation (MIFO) has also been proposed [19]:

$$\min_{\beta,\, c,\, s} \; \frac{\sum_{i=1}^{N} \left(y_i - c - \sum_{j=1}^{P} \beta_j x_{ij} \right)^2}{s} \tag{8a}$$

$$\text{s.t. } s \leq \sum_{i=0}^{P} g(i) w_i \tag{8b}$$

$$\sum_{i=0}^{P} i \cdot w_i = \sum_{i=1}^{P} z_i \tag{8c}$$

$$\sum_{i=0}^{P} w_i = 1 \tag{8d}$$

$$-Mz \leq \beta \leq Mz \tag{8e}$$

$$z \in \{0,1\}^P,\ w \in \{0,1\}^{P+1},\ \beta \in \mathbb{R}^P,\ s \geq 0, \tag{8f}$$

where s is an additional variable which makes the objective function convex and $g(x) = \exp(-\alpha x/N)$. Problem (8) can be tackled exploiting the efficient MIQO solvers. In particular, the parametrized problems

$$d(t) = \min_{\beta,\ c,\ s} R(\beta,\ c) - ts \tag{9a}$$

$$\text{s.t. (8b) – (8f)} \tag{9b}$$

are considered. If $d(t^\star) = 0$, then $t^\star$ is the optimal value of (8). The original problem is thus solved finding a root of the equation $d(t) = 0$ by using Newton method. The whole procedure thus reduces to solving a sequence of problems of the form (9), which can be proved to terminate at most after the $P + 1$-th problem.

3 The Proposed Method

Let us consider optimization problems of a slightly more general form w.r.t. (4):

$$\begin{aligned} \min_{\beta,c,\sigma} \quad & N\log(\sigma^2) + \frac{1}{\sigma^2}R(\beta,c) + g(\beta) \\ \text{s.t.} \quad & \sigma > 0, \quad c \in \mathbb{R} \\ & \beta \in \mathbb{R}^P, \end{aligned} \tag{10}$$

where $g : \mathbb{R}^P \to S$, being S a discrete set with finite cardinality, and $R : \mathbb{R}^P \times \mathbb{R} \to (\epsilon, +\infty)$ is a quadratic convex function. This model formulation is justified by linear regression, as we will shortly see; the model might be enriched by additional constraints, as we will do in section 4 when dealing with autoregressive models. The addition of those constraints does not change the properties presented here.

In this paper, we propose a new method to solve such type of problems. Note that problem (4) is indeed an instance of (10), where

$$g(\beta) = \alpha\|\beta\|_0, \qquad R(\beta, c) = \sum_{i=1}^{N}\left(y_i - c - \sum_{j=1}^{P}\beta_j x_{ij}\right)^2.$$

We assume that R takes values in $(\epsilon, +\infty)$, with $\epsilon > 0$. This assumption is not only reasonable in practice, but also necessary from a theoretical perspective: in linear regression, if a linear model perfectly fits the data, the variance σ^2 of the residuals becomes 0 and thus the likelihood is no more defined. More in general, the objective function, for any fixed β and c, admits a finite optimum only if $R(\beta, c) > 0$.

Fixing $\bar{\beta}, \bar{c}$, we get the convex, univariate problem

$$\min_{\sigma>0} \ N\log(\sigma^2) + \frac{R(\bar{\beta}, \bar{c})}{\sigma^2},$$

whose closed form solution, if $R(\bar{\beta}, \bar{c}) > 0$, is immediately found to be

$$\bar{\sigma}^2 = \frac{R(\bar{\beta}, \bar{c})}{N}. \tag{11}$$

Note that this relation has to hold at every optimal solution $\bar{\beta}, \bar{c}, \bar{\sigma}$ of problem (10), so we might equivalently reformulate the problem as

$$\min_{\beta,c} N \log\left(\frac{R(\beta,c)}{N}\right) + g(\beta) + N, \tag{12}$$

similarly as in (6).

However, differently from e.g. the MISOCP [29] or the MIFO [19] formulations, we don't employ formulation (12). On the contrary, we propose a two blocks Gauss-Seidel type solving scheme [5] for formulation (10). Our approach, as we will see later, is capable to produce a much more efficient method. The procedure is described by Algorithm 1.

Algorithm 1 Alternate Minimization (AM)

Input: $\beta^0,\ c^0,\ \sigma_0,\ k = 0$

1: let $g(\beta^{-1}) = \texttt{NaN}$

2: **while** $g(\beta^k) \neq g(\beta^{k-1})$ **do**

3: set

$$\beta^{k+1},\ c^{k+1} = \arg\min_{\beta,\ c} \frac{R(\beta,c)}{{\sigma_k}^2} + g(\beta)$$

4: set

$$\sigma^2_{k+1} = \arg\min_{\sigma^2>0} N\log(\sigma^2) + \frac{R(\beta^{k+1},\ c^{k+1})}{\sigma^2} = \frac{R(\beta^{k+1},\ c^{k+1})}{N}$$

5: set $k = k + 1$

6: **end while**

7: **return** $\beta^k,\ c^k,\ \sigma_k$

The major computational effort required by Algorithm 1 resides in the solution of the optimization problems defined at step 3 of the procedure.

At first glance, dealing with such subproblems may seem particularly hard, since the algorithm requires the global optimum and the objective function is discontinuous. However, problem

$$\min_{\beta,\ c} \quad \frac{R(\beta,c)}{{\sigma_k}^2} + g(\beta) \tag{13}$$

can be equivalently reformulated as a mixed-integer convex quadratic problem; in particular it is equivalent to:

$$\min_{\beta,\ c} \ R(\beta,\ c) + \Lambda\sigma_k^2 \tag{14a}$$

$$\text{s.t.} \ \ \Lambda = g(\beta) \tag{14b}$$

$$\Lambda \in S, \tag{14c}$$

$$\beta \in \mathbb{R}^P, \quad c \in \mathbb{R}, \tag{14d}$$

where, based on the structure of g, the constraint (14b) can be turned into a linear constraint by introducing auxiliary binary and integer variables and linear constraints. For example, if $g(\beta) = \alpha\|\beta\|_0$, as in the case of linear regression, we

can substitute constraints (14b)-(14c) with the following set of constraints:

$$-M\delta_i \leq \beta_i \leq M\delta_i \quad \forall\, i = 1, \ldots, P \tag{15a}$$

$$\Lambda = \alpha \sum_{i=1}^{P} \delta_i. \tag{15b}$$

If M is a large enough positive constant, the value of variable Λ, appearing in the objective function, will be equal to the number of non-zero components of β: indeed, if $\beta_i \neq 0$, then δ_i will have to assume the value 1 in order to satisfy constraint (15a), while if $\beta_i = 0$, then δ_i, which may assume both values 0 and 1, will be 0, since it brings to a lower objective value than if it takes 1.

Problem (14) with constraints of the type (15a)-(15b) is usually solved significantly faster than, e.g., the MISOCP problem (7) that directly tackles problem (12) [29]. Indeed, solving a sequence of problems of form (14) proved to be more efficient than solving (7), as we will show in Section 5.

In fact, problem (14) is similar, in terms of complexity, to problem (3), which is solved P times through enumeration. On the other hand, even though we will briefly proof that Algorithm 1 also solves, in the worst cases, P times problem (14), it in fact usually stops after much less than P iterations.

The following Proposition characterizes the properties of Algorithm 1. In order to simplify the notation throughout the analysis, we will ignore, with no loss of generality, the intercept term c. Before going on with the analysis, we also make an assumption, which is reasonable in practical implementations, about Algorithm 1:

Assumption 1 *Step 3 of Algorithm 1 is performed in such a way that if* $R(\beta^{k+1})/\sigma_k^2 + g(\beta^{k+1}) = R(\beta^k)/\sigma_k^2 + g(\beta^k)$ *then* $\beta^{k+1} = \beta^k$.

The above assumption substantially says that the current point is updated only if the new point is strictly better than the previous one in terms of objective value. For the sake of notation simplicity, let also $f(\beta) = N \log(R(\beta)/N) + g(\beta)$. Now, we can finally turn to the convergence analysis.

Proposition 1 *Consider Algorithm 1, under Assumption 1. Then, the following properties hold:*

(a) For each iteration k, *either* $g(\beta^k) = g(\beta^{k-1})$, *i.e. the algorithm terminates, or* $g(\beta^k) \neq g(\beta^h)$ *for all* $h < k$.

(b) The algorithm terminates in at most $|S|$ *iterations, returning a solution* $(\bar{\beta}, \bar{\sigma})$.

(c) Let $\bar{k}$ *be the index of the last iteration. Then* $f(\bar{\beta}) \leq f(\beta)$ *for all* $\beta \in \{\beta \mid \exists\, k \in \{1, \ldots, \bar{k}\}$ *s.t.* $g(\beta^k) = g(\beta)\}$

(d) If $\bar{k} = |S| + 1$, *then the returned solution* $\bar{\beta}$ *is optimal.*

(e) Let $\beta^\star$ *be an optimal solution of problem* (12), *i.e., the pair* $\beta^\star$, $\sigma^\star = R(\beta^\star)/N$ *is optimal for problem* (10). *Then, the following bound holds:*

$$0 \leq f(\bar{\beta}) - f(\beta^\star) \leq -N \log(1 - \eta^2 \exp(\theta - 1)), \tag{16}$$

where $\theta \in (0, 1)$ *and* $\eta = (g(\bar{\beta}) - g(\beta^\star))/N$

Proof We prove one property at a time.

(a) Since both step 3 and step 4 require to compute global minima of subproblems, the sequence of objective values $\{f(\beta^k)\}$ is monotone non-increasing.

By the instructions of the algorithm,

$$\beta^{k+1} = \arg\min_{\beta} \frac{R(\beta)}{\sigma_k^2} + g(\beta),$$

so it also holds

$$\beta^{k+1} = \arg\min_{\beta\,:\,g(\beta)=g(\beta^{k+1})} \frac{R(\beta)}{\sigma_k^2} + g(\beta) = \arg\min_{\beta\,:\,g(\beta)=g(\beta^{k+1})} R(\beta). \tag{17}$$

Therefore, if $g(\beta^h) = g(\beta^k)$ for two indexes h and k, $R(\beta^k) = R(\beta^h)$ and thus $f(\beta^k) = f(\beta^h)$. Now, let $k > h$ and assume $g(\beta^k) = g(\beta^h)$. Since the sequence of objective values is decreasing, we have

$$f(\beta^k) \leq f(\beta^\ell) \leq f(\beta^h) = f(\beta^k)$$

for all $k > \ell \geq h$, where the last equality comes from the previous considerations. Thus, we have

$$f(\beta^\ell) = f(\beta^k)$$

for all $k > \ell \geq h$.

Since the objective value has not decreased through iterations h to k, from Assumption 1 it has to be $\beta^\ell = \beta^h$ for all $k > \ell \geq h$.

But then, $g(\beta^\ell) = g(\beta^h)$ for all $k > \ell \geq h$, which is only possible if $\ell = h = k - 1$, since $g(\beta^{h+1}) = g(\beta^h)$ triggers the stopping criterion.

Therefore, for all $h < k$, we have either $h = k - 1$ with $g(\beta^k) = g(\beta^{k-1})$ or $g(\beta^k) \neq g(\beta^h)$.

(b) Since g can have at most $|S|$ possible different values and, from (a), at each iteration the algorithm either finds a solution with an unseen value of g or maintains the previous solution, no later than at the beginning of the $|S|+1$-th iteration the stopping criterion fires.

(c) From the non-increasing property of $\{f(\beta^k)\}$, $f(\beta^{\bar{k}}) \leq f(\beta^k)$ for all k. Moreover, recalling that, for all k, (17) holds, it has to be

$$\beta^k = \arg\min_{\beta\,:\,g(\beta)=g(\beta^k)} f(\beta) = N \log\left(\frac{R(\beta)}{N}\right) + g(\beta).$$

Thus we can conclude that

$$\beta^{\bar{k}} = \arg\min_{\beta\,:\,\exists\, h\in\{1,\ldots,\bar{k}\}\,:\,g(\beta)=g(\beta^h)} f(\beta).$$

(d) This property directly descends from (a), (b) and (c).

(e) Since $\beta^\star$ is the optimal solution, we have

$$\beta^\star = \arg\min_{\beta} N \log\left(\frac{R(\beta)}{N}\right) + g(\beta), \tag{18}$$

while, by the instructions of the algorithm, we know that the returned solution $(\bar{\beta}, \bar{\sigma})$ satisfies

$$\bar{\sigma}^2 = \frac{R(\bar{\beta})}{N}, \tag{19}$$

$$\bar{\beta} = \arg\min_{\beta} N\frac{R(\beta)}{R(\bar{\beta})} + g(\beta). \tag{20}$$

From (18), we have

$$N \log\left(\frac{R(\beta^\star)}{N}\right) + g(\beta^\star) \leq N \log\left(\frac{R(\bar{\beta})}{N}\right) + g(\bar{\beta}),$$

while (20) implies

$$N + g(\bar{\beta}) \leq N\frac{R(\beta^\star)}{R(\bar{\beta})} + g(\beta^\star).$$

Now, let $r = R(\beta^\star)/R(\bar{\beta})$ and $\Delta = g(\bar{\beta}) - g(\beta^\star)$. We can rewrite the previous inequalities in a more compact way:

$$N + \Delta \leq Nr \quad \Longleftrightarrow \quad r \geq 1 + \Delta/N \tag{21}$$

$$N \log(r) \leq \Delta \quad \Longleftrightarrow \quad r \leq \exp(\Delta/N) \tag{22}$$

Let us recall Taylor's theorem:

$$s(y) = s(x) + s'(x)(y-x) + s''(\xi)(y-x)^2, \quad \xi = x + \theta(y-x) \text{ for some } \theta \in (0,1),$$

and let $\eta = \Delta/N$. Setting $s(\eta) = \exp(\eta)$ and $x = 0$, we can write $\exp(\eta) = 1 + \eta + \eta^2 \exp(\theta\eta)$, for some $\theta \in (0,1)$. Rearranging we get

$$1 + \eta = \exp(\eta) - \eta^2 \exp(\theta\eta).$$

Combining inequalities (21) and (22) with the last equality, we obtain the following bounds for r:

$$\exp(\eta) - \eta^2 \exp(\theta\eta) \leq r \leq \exp(\eta).$$

Now, let us consider the gap, in terms of objective value, between the returned solution $\bar{\beta}$ and the optimal solution $\beta^\star$; we have

$$\begin{aligned}
0 \leq f(\bar{\beta}) - f(\beta^\star) &= N \log(1/r) + \Delta \\
&= \Delta - N \log(r) \\
&\leq \Delta - N \log(\exp(\eta) - \eta^2 \exp(\theta\eta)) \\
&= \Delta - N \log(\exp(\eta)(1 - \eta^2 \exp(\eta(\theta - 1)))) \\
&= \Delta - N \log(\exp(\eta)) - N \log(1 - \eta^2 \exp(\theta - 1)) \\
&= \Delta - N(\Delta/N) - N \log(1 - \eta^2 \exp(\theta - 1)) \\
&= -N \log(1 - \eta^2 \exp(\theta - 1)),
\end{aligned}$$

i.e., we have obtained the bound of property (e).

Note that the upper bound at point (e) of Proposition 1 is often very close to 0. Consider for example the case of linear regression, where $g(\beta) = \alpha\|\beta\|_0$. The bound becomes

$$f(\bar{\beta}) - f(\beta^\star) \leq -N \log\left(1 - \frac{\alpha^2(\|\bar{\beta}\|_0 - \|\beta^\star\|_0)^2}{N^2} \exp(\theta - 1)\right);$$

the bound goes fast to zero when $\alpha(\|\bar{\beta}\|_0 - \|\beta^\star\|_0)/N$ goes to zero. Therefore:

- the gap is zero if $\|\bar{\beta}\|_0 = \|\beta^\star\|_0$;
- if $\alpha = 0$, there is no penalty on the model complexity; $\bar{\beta}$ then minimizes $R(\beta)$ and is trivially the global optimum;
- if the absolute value of $\alpha(\|\bar{\beta}\|_0 - \|\beta^\star\|_0)/N$ is small, the optimality gap is bounded by a small quantity; moreover, note that $\left|\|\bar{\beta}\|_0 - \|\beta^\star\|_0\right| \leq P$; in most applications $P \ll N$ and for the GOF measures we have considered it holds that $\alpha = o(N)$, so we are typically guaranteed to obtain, at least, a nearly-optimal solution.

Note that optimality of the returned solution $\bar{\beta}$ cannot be guaranteed. In fact, we can show by a numerical counter-example that, in unfortunate cases, Algorithm 1 may stop at suboptimal solutions:

Example 1 Consider the best subset selection problem for linear regression, using AIC as GOF measure, for the model

$$Y = X\beta,$$

where

$$Y = \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} \qquad X = \begin{bmatrix} 10 & 0.1 \\ 0.1 & 10 \\ 1 & 1 \end{bmatrix}$$

and $\beta \in \mathbb{R}^2$ (no intercept term c). The AIC measure for the model as a function of β is given, up to constants, by

$$f(\beta) = 3 \log\left(\frac{\|Y - X\beta\|_2^2}{3}\right) + 2\|\beta\|_0.$$

The optimal parameters $\beta^\star$ for this problem are

$$\beta^\star = \begin{bmatrix} 1.0673 \\ 1.0673 \end{bmatrix},$$

with $f(\beta^\star) = 13.74$. However, if we run Algorithm 1 setting $\sigma_0 = 178.0219/3 = 59.34$, it stops at the end of the second iteration returning the first solution found:

$$\bar{\beta} = \begin{bmatrix} 1.0989 \\ 0 \end{bmatrix},$$

having value $f(\bar{\beta}) = 14.25 > f(\beta^\star)$.

Indeed, let β_0, $\beta_1 = \bar{\beta}$ and $\beta_2 = \beta^\star$ the solutions with L_0 norm respectively equal to 0, 1 and 2 minimizing $R(\beta) = \|Y - X\beta\|_2^2$. We have $R(\beta_0) = 300$, $R(\beta_1) = 178.022$ and $R(\beta_2) = 63.08$. With $\sigma_0 = 59.34$, the objective values of these solutions in the problem at step 3 of Algorithm 1 are respectively 5.056, 5 and 5.063: β_1 is therefore optimal; but then,

the value of σ_1 is set to $R(\beta_1/3) = 59.34$, i.e. it doesn't change w.r.t. σ_0; thus, the first iteration is exactly repeated and at the end of it the stopping condition gets true, with the algorithm returning $\beta_1 = \bar{\beta}$.

It's interesting noting that the gap in terms of objective value is not so narrow in this case. However, this simple example doesn't match with the typical real-world problems where $P \ll N$.

4 Application to Time Series

A time series $\{X_t\}$ is generated by a stationary AR(p) process with Gaussian error terms if the following holds:

$$\begin{gathered} X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t, \\ \varepsilon_t \sim \text{i.i.d. } \mathcal{N}(0, \sigma^2), \end{gathered} \tag{23}$$

where $c \in \mathbb{R}$ is the intercept term, $\varphi \in \mathbb{R}^p$ is the vector of autoregressive parameters which explain the temporal dynamics of the time series and ε_t, for $t = 1, 2, \ldots$, are Gaussian white-noise terms with variance σ^2.

In many applications finite-order autoregressive models of time series are useful for prediction and inference [36]. The interest usually lies in stationary AR models. A stochastic process is (weakly) stationary if its mean and autocovariance are invariant w.r.t. time and the variance is always finite:

$$\begin{aligned} E(X_{t_1}) &= E(X_{t_2}) && \forall\, t_1, t_2 \\ Cov\,(X_{t_1}, X_{t_2}) &= Cov\,(X_{t_2 - t_1}, X_0) && \forall\, t_2 \geq t_1 \geq 0 \\ E\left(X_t^2\right) &< \infty && \forall\, t \end{aligned}$$

It can be proven [21] that a process generated according to (23) is stationary if and only if the autoregressive coefficients $(\varphi_1, \ldots, \varphi_p) \in \mathbb{R}^p$ are such that the roots of the associated characteristic equation

$$\pi_\varphi(z) = 1 - \varphi_1 z - \varphi_2 z^2 - \cdots - \varphi_p z^p = 0, \tag{24}$$

lie outside the unit complex circle.

Given a time series of size N generated from (23), we immediately derive the same formulation of a linear regression model by setting in (1) $\beta = \varphi$, $y_i = X_i$ and $x_{ij} = X_{i-j}$.

However, some issues arise when fitting autoregressive models on given data, w.r.t. linear regression models:

- In linear regression, the number of parameters is a priori known; subset selection is a refining, optional step. On the other hand, with AR models the number of parameters is typically not available in advance and subset selection is therefore required.
- Subset selection in AR models is slightly different w.r.t. the linear regression case, as in autoregression we deal with the model order. An AR model has order p (it is an AR(p) model) if the current sample only depends on the previous p samples. As a consequence, when comparing two AR models of different order, penalty terms on the model complexity are proportional to the order of the model, not to the number of non-zero variables.

– With autoregressive time-series, stationarity constraints have to be taken into account.

– If the autoregressive model is cast into the linear regression equation, elements x_{ij} of matrix X are undefined for $i - j < 0$. Consequently, if one performs least squares regression on the AR series, the first p samples of the series have to be ignored. More formally, let us consider a maximum likelihood estimate. The (log) likelihood function for AR models is more complicated than that of linear regression, since it should take into account the joint distribution of the first samples. In particular, under stationarity assumptions, it takes the form [21]

$$-2\ell(\varphi, c, \sigma^2) = N\log(2\pi) + N\log(\sigma^2) - \log|V_p^{-1}| + \\ + \frac{1}{\sigma^2}(\bar{X}_p - \mu_p)^T V_p^{-1}(\bar{X}_p - \mu_p) + \frac{1}{\sigma^2}\sum_{t=p+1}^{N}\left(X_t - c - \sum_{i=1}^{p}\varphi_i X_{t-i}\right)^2 \tag{25}$$

where:

- $(V_p^{-1})_{i,j} = \sum_{k=0}^{i-1}\varphi_k\varphi_{k+j-i} - \sum_{k=p+1-j}^{p+i-j}\varphi_k\varphi_{k+j-i}$ for $1 \le i \le j \le p$; values of V_p^{-1} with $j > i$ can be inferred from the fact that the matrix is symmetric;
- $\bar{X}_p = (X_1, \ldots, X_p)^T \in \mathbb{R}^p$ is the vector of the first p samples of the series;
- $\mu_p = \mu\, e \in \mathbb{R}^p$, with $\mu = c/(1 - \varphi_1 - \cdots - \varphi_p)$.

If we consider the conditional likelihood, i.e., the likelihood conditioned to the value of the first p samples, we actually get the likelihood of a linear regression model. The conditional likelihood is in fact a good approximation of the exact likelihood, especially when $p \ll N$, and is therefore often used as a surrogate when doing maximum likelihood estimation or comparisons of models by means of GOF measures like AIC, BIC and HQIC. In particular, the exact maximum likelihood estimates and the conditional ones have the same large sample distribution [21].

Once the order of the AR model has been selected, all the parameters of the model $(\varphi_1, \varphi_2, \ldots, \varphi_p, c, \sigma^2)$ are usually estimated by maximum likelihood [21] or by the method of moments solving the Yule and Walker equations [10]. In this scenario, the selection of the AR order, i.e. the appropriate number of past values that affect the present, represents one of the main issues.

The Box and Jenkins procedure [10,21] suggests the identification of the order through the analysis of the estimated autocorrelation function (ACF) and the partial autocorrelation function (PACF). This procedure however is highly subjective and requires statistical knowledge. Moreover, it is not feasible when several AR models have to be fitted automatically, in an on line, real time system.

Thus, there is a need for automatic algorithms that perform model selection and fitting in the class of autoregressive models. These algorithms have to base the evaluation of models of different orders on GOF measures such as AIC, BIC or HQIC. Therefore, taking into account all previous considerations, the problem that is solved can be formalized as follows

$$\min_{\varphi,\, c,\, \sigma^2} \; -2\ell(\varphi,\, c,\, \sigma^2) + \alpha\, \mathrm{ord}(\varphi) \tag{26a}$$

$$\mathrm{ord}(\varphi) = \max\{j = 0, \ldots, P \mid \varphi_j \neq 0\}, \;\; \varphi_0 = 1 \tag{26b}$$

$$\varphi \text{ satisfies stationarity constraints} \tag{26c}$$

$$\varphi \in \mathbb{R}^P,\; c \in \mathbb{R},\; \sigma^2 \in \mathbb{R}_+, \tag{26d}$$

where P is a pre-set upper bound on the model order. The state-of-the-art methods to solve this problem, like the one implemented in function `auto.arima()` from the popular `R` library `forecast` [23], employ enumerative-like strategies: for various (possibly all) values of $p \in \{0, P\}$, the order of the AR model is fixed; for each of these values of p, the best AR(p) model is estimated, e.g. by maximum likelihood. Then, the GOF measures of these models are compared in order to chose the optimal one.

Now, there are some issues that need to be handled and options to be selected:

- The search over the orders can be performed exhaustively or in a step-wise fashion; this consists in a greedy procedure which, starting from a tentative order p, explores its close neighbors $p + 1$ (forward) or $p - 1$ (backward) or both, stopping when no improvement is observed. Clearly, the step-wise scheme is faster than exhaustive search, but might provide worse solutions.
- In the model selection phase, the best model for a given order can be identified according to the exact likelihood or the conditional likelihood; again, the approximate way is faster, while the exact minimization may result in more accurate solutions.

  Note that the exhaustive search coupled with the minimization of the exact likelihood provides the optimal solution, except if stationarity issues arise (see the next point).

  On the other hand, the model selection phase, when conducted through conditional likelihood minimization, can be followed by a refinement step, i.e. a local search on the exact likelihood, with fixed order p, starting from the point minimizing the conditional likelihood.
- Handling stationarity constraints is not trivial. For many algorithms, if in the model selection step the optimal solution for some order p is non-stationary, then that order is discarded altogether, even though the optimal model might nonetheless be found within models of that order.

The algorithm proposed in Section 3 can be adapted to the case of problem (26). The additions with respect to Algorithm 1 that we have introduced are:

- We consider the conditional likelihood; at the end of the entire process, we perform a refinement local search based on the exact likelihood, having fixed the order.
- We set $g(\beta) = \alpha\, \mathrm{ord}(\varphi)$, as in (26b).

– The stationarity constraints can be written in closed form if $p = 1$ or $p = 2$. In particular, when $p = 1$, stationarity condition holds if and only if $|\varphi_1| < 1$; when $p = 2$, it holds if and only if $|\varphi_1| < 1 - \varphi_2$ and $|\varphi_2| < 1$ [21].

These constraints can be added to (15a) and (15b) as follows. We introduce binary indicator variables z_j that are 1 if and only if $\Lambda = \alpha\, j$; this is modeled by

$$z_j \in \{0, 1\},\ j = 0, \ldots, P \tag{27a}$$

$$\sum_{j=0}^{P} z_j = 1 \tag{27b}$$

$$\alpha \sum_{j=0}^{P} j z_j = \Lambda. \tag{27c}$$

Then, the stationarity constraint for first order model is modeled by,

$$-(z_1 + M(1 - z_1)) \leq \varphi_1 \leq (z_1 + M(1 - z_1)) \tag{28}$$

while the constraints for second order models are

$$\varphi_1 + \varphi_2 \leq z_2 + M(1 - z_2) \tag{29a}$$

$$\varphi_2 - \varphi_1 \leq z_2 + M(1 - z_2). \tag{29b}$$

As for orders $p > 2$, at step (3) of Algorithm 1 we reject the new solution if the stationarity test is not satisfied and add a constraint to prohibit solutions of order p ($\Lambda \neq \alpha\, p$). As a consequence, our algorithm suffers from the same problem of possibly cutting out good solutions, but at least we prevent this glitch for models of order 1 and 2.

Last, if the final local search produces a non-stationary solution, then the stationary starting solution is retained.

The procedure is summarized by Algorithm 2.

Algorithm 2 Alternate Minimization for Autoregressive Models

Input: $\varphi^0,\ c^0,\ \sigma_0,\ k = 0$

let $\bar{g} = -1$

while $g(\varphi^k) \neq \bar{g}$ **do**

set

$$\varphi^{k+1},\ c^{k+1} = \underset{\varphi,\ c}{\arg\min}\ \frac{R(\beta, c)}{{\sigma_k}^2} + g(\beta)$$

$$\text{s.t. } \varphi \text{ satisfies stationarity constraints if } g(\varphi) = 1 \text{ or } g(\varphi) = 2$$

$\bar{g} = g(\varphi^{k+1})$

if $\pi_{\varphi^{k+1}}(z)$ has any root inside the unit circle **then**

set $\varphi^{k+1} = \varphi^k$

add the constraint $g(\varphi) \neq \bar{g}$ to the model

end if

set

$$\sigma^2_{k+1} = \frac{R(\varphi^{k+1},\ c^{k+1})}{N}$$

set $k = k + 1$

end while

return $\varphi^k,\ c^k,\ \sigma_k$

5 Computational Experiments

5.1 Preparation

Concerning linear regression, similarly as in previous works on the topic [29,24], our benchmark is made up of eight datasets from the UCI Machine Learning Repository [16]. Table 1 synthetically describes these datasets, showing the number N of points and the number P of variables. The Solar Flare instance has three target variables (C, M and X) and therefore three instances of the problem have been prepared.

As for the Forest Fires dataset, interaction terms between x and y spatial coordinates have been created. For each dataset we performed the one-hot encoding for the categorical variables and we normalized the other ones to zero mean and unit standard deviation, in order to prevent numerical issues. Moreover, data points with missing variables haves been removed.

Minimization of AIC, BIC and HQIC for the 10 problems has been carried out. With all the considered methods in the experiments, Gurobi 8.1.0 [20] was employed as the quadratic programming solver for MINLP problems and subproblems. Indeed, all of the subproblems we are solving with our approach are Mixed Integer Quadratic programming problems (MIQP), i.e. models with quadratic objective, linear constraints and both continuous as well as discrete variables. Although still complex in general, these models can be successfully solved in the convex case through general purpose

Dataset	N	P
Housing	506	13
Servo	167	19
Auto MPG	392	25
Solar Flare C	1066	27
Solar Flare M	1066	27
Solar Flare X	1066	27
Breast Cancer Wisconsin	194	33
Forest Fires	517	63
Automobile	159	65
Communities and Crime	1993	102

Table 1: List of datasets for experiments on subset selection for linear regression.

solvers like Gurobi if the dimension is not excessively large. In order to enhance the efficiency of Gurobi, continuous variables were constrained to belong to the interval $[-10^3, 10^3]$. Being the datasets normalized, this in practice doesn't represent a restriction to the model. In order to prevent numerical problems with integer variables, we set the integer precision parameter to 10^{-9}, which is the most accurate possible value with Gurobi. The value of M for big$M$-type constraints was set to 10^4.

Regarding the case of autoregressive models, we considered real and synthetic datasets. Concerning the real time series, we considered a dataset of 10 series of various origin and length normalized to zero mean and unit variance. This collection of series includes both seasonal and non-seasonal series, whose main characteristics are summarized in Table 2.

Autoregressive models are often applied to series of general form after preprocessing steps have been carried out in order to handle possible non-stationarity issues, caused, e.g., by seasonality or trends. Regardless of how stationarity is obtained, autoregressive models are then a suitable choice, due to the presence of autocorrelation between the elements of the processed time series.

A pre-processing phase to obtain level-stationarity (i.e. constant mean and variance w.r.t. time) was thus performed for all time series. With non-seasonal time series, we removed the possible presence of a trend by differencing or detrending, depending on the presence of unit roots. Differencing is necessary in presence of unit roots. The unit roots test is performed by means of KPSS test [26]. The required number of period-to-period differences is selected through successive KPSS tests: if the KPSS test statistic is significant (i.e., a unit root is likely to be present), the differenced time series is tested, iterating the process.

When the KPSS test statistic is no more significant, the detrending strategy, carried out by least squares, is sufficient to obtain a stationary time series: residuals of the estimated model constitute the transformed stationary time series.

As for seasonality, if the series has no trend, stationarity is achieved computing seasonal differences.

Series	Source	Length N	Trend	Seasonality
Annual real gnp Usa (1909-1970)	DataMarket	62	Yes	No
Internet traffic data	Github	14773	No	Yes
Number of earthquakes per year	DataMarket	99	No	No
CO2 (ppm) mauna loa	DataMarket	192	Yes	Yes
Real daily wages in pounds, England	DataMarket	735	Yes	No
Air Passengers	DataMarket	144	Yes	Yes
Annual changes in the earth's rotation	DataMarket	150	No	No
Quarterly gross fixed capital expenditure, Australia	DataMarket	144	Yes	Yes
Quarterly Iowa nonfarm income	DataMarket	128	Yes	No
Monthly mean thickness ozone column Arosa	DataMarket	518	No	Yes

Table 2: List of time series for experiments on subset selection for autoregression.

The number of seasonal differences sufficient to obtain a stable seasonality pattern is selected through successive OCSB tests [32]. If, on the other hand, both trend and seasonality are present, the procedure changes based on the type of seasonality. When the seasonality pattern is stable over time, a linear model with dummy seasonal variables is fitted by least squares. With non-stable seasonality, seasonal differencing is combined together with period-to-period differencing in the presence of a seasonal unit root.

Lastly, it is worth noting that for the *internet traffic data* time series, because of the presence of a multiple seasonality pattern, we obtain a stationary time series from residuals of a TBATS model setting a double seasonal period [14].

The synthetic benchmark consists of 400 autoregressive series of length $N \in \{100, 200, 500, 1\,000\}$ and order $p \in \{1, 2, 3, 4, 5\}$ (20 series for each combination of N and p). For each of these series, the coefficients $\varphi_1, \ldots, \varphi_p$ were randomly sampled from a Gaussian distribution and, after stationarity tests, the series samples were produced with the `R` function `arima.sim()` from `stats` package [37]. This function generated the series according to equation (23), by setting $c = 0$ and $\sigma \in \{5, 10, 50, 100\}$ (5 series for each combination of σ, p and N).

We carried out the optimization of AIC, BIC and HQIC with various solvers on both the real and the synthetic benchmark. Gurobi was again used as MINLP solver, with analogous setup as for the linear regression case.

All experiments were performed on a machine with Ubuntu Server 16.04 LTS OS, Intel Xeon E5-2430 v2 @ 2.50GHz CPU and 16GB RAM. Every run of each algorithm was performed on a single CPU core, in order to make time measurements fair, and a time limit of 10 000 seconds was set.

In the following, we will make use of performance profiles [15] for comparing the performance of different algorithms on a set of benchmark problems. We recall that, in performance profiles, each curve represents the cumulative fraction of problems the corresponding solver could solve in a time which is at most a factor of τ worse than the best performing one. Separate analyses are performed for different quality metrics.

5.2 Linear Regression

In our experiment, we compared several solvers on the subset selection for linear regression problems generated from datasets in Table 1 using AIC, BIC and HQIC as GOF measures. We considered our Alternate Minimization (AM) algorithm (Algorithm 1), the step-wise heuristic with Forward selection and Backward elimination strategies, the exhaustive approach, the MISOCP model (7), the Newton's method to solve problem 9 (MIFO-Newton) and finally the method of solving the MIQP model (3) for all possible values of k.

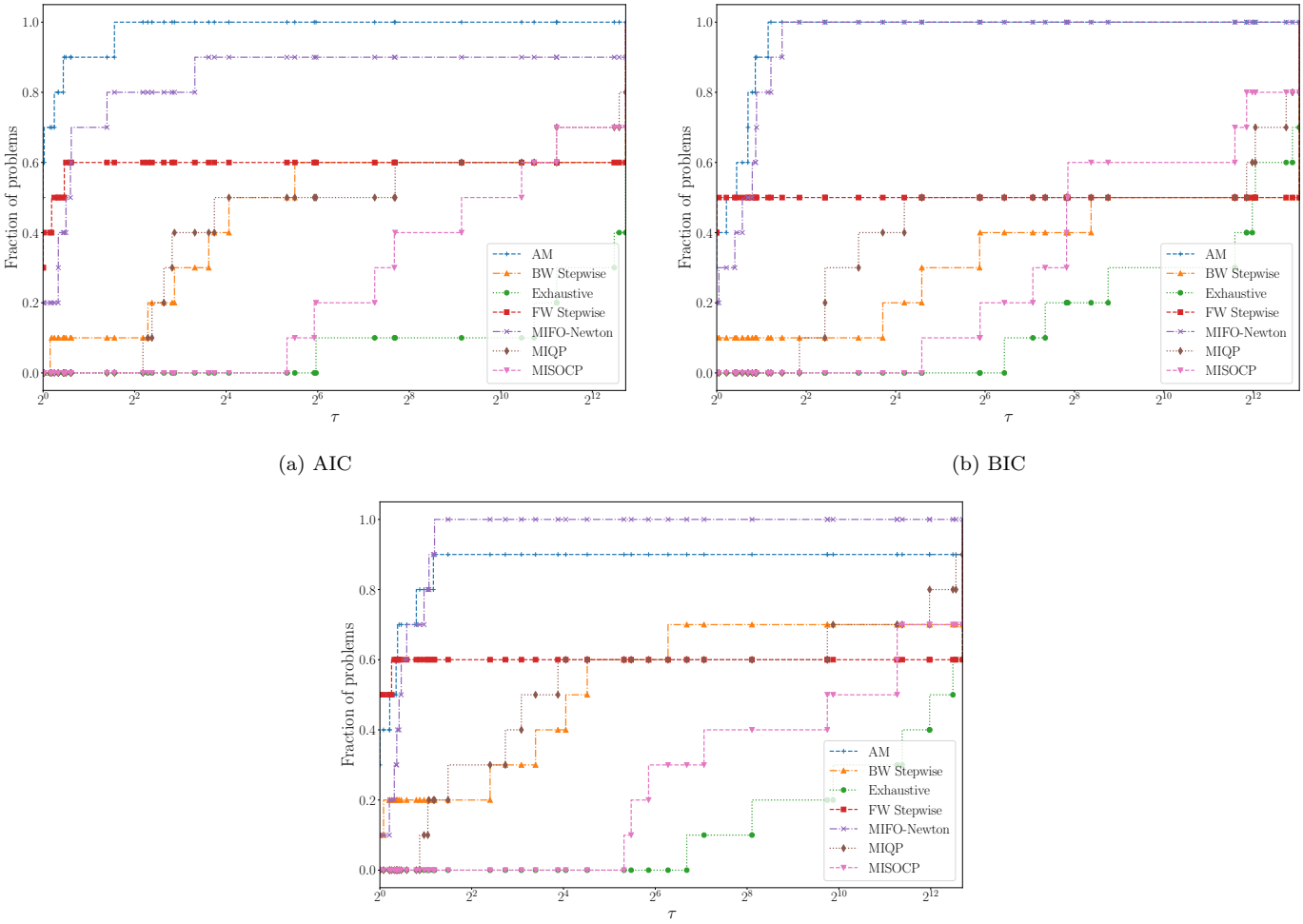
AM, MIFO-Newton and the iterated MIQP algorithms employ mixed-integer solvers (Gurobi) as subroutines. With the largest problems (those generated from Automobile and Communities and Crime datasets) we had to set an inner time limit of 1 200 seconds for each iteration, since otherwise Gurobi never stops, failing at certificating the optimality of the current solution of the subproblem. This slight modification clearly spoils the theoretical properties of the algorithms, but in practice leads to good performance both in terms of runtime and quality of the solutions.

A warm-start strategy also speeds up algorithms AM and MIFO-Newton. The solution found at the k-th iteration is feasible (and likely good) for the $k+1$-th problem; using that solution as starting point at the $k+1$-th iteration provides some reduction in the computing time. On the other hand, the addition of redundant constraints based on normal equations, as outlined in [19], greatly improves the performance of MISOCP model. The same addition, on the contrary, turned out to be quite useless with the other mixed-integer models.

Method	# successful runs	total time (sec)
AM	29/30	31299
BW Stepwise	18/30	38666
Exhaustive	17/30	267405
FW Stepwise	17/30	3530
MIFO-Newton	29/30	32759
MIQP	25/30	155295
MISOCP	22/30	172215

Table 3: For each considered algorithm, the number of times the optimal solution was found and the sum of runtimes is reported, out of the 30 linear regression problems (10 datasets with AIC/BIC/HQIC). A solution is considered optimal if the relative distance to the best overall objective value is lower than 10^{-3}.

In Table 3 we show the overall performance of the algorithms. We can see that the stepwise heuristics are fast, especially the forward selection, but often lead to suboptimal solutions. The exhaustive search is very slow, and often exceeds the time limit, returning bad solutions. The MISOCP model has a slightly better behavior, but with similar shortcomings. Our proposed AM algorithm proved to be the best choice, as it is the second one in terms of CPU time



(a) AIC

(b) BIC

(c) HQIC

Fig. 1: Performance profiles of runtimes of different algorithms for subset selection in 10 linear regression problems. The problems are generated from datasets in Table 1. Different GOF measures are considered.

while being the top ranking in terms of the number of solved problems; like MIFO-Newton, it only fails once at finding an optimal solution, but it requires a smaller time to run on all the problems.

Figures 1 and 2 give a wider insight of tests results. In Figure 1 the performance profiles, in terms of runtime, of the seven considered algorithms on the ten regression problems are shown. Note that we considered the runtime to be "infinite" when the returned solution is suboptimal. We made up separate profiles for AIC (1a), BIC (1b) and HQIC (1c). We can observe that the profile of AM is almost completely dominant with respect to the others. The only comparable algorithm is MIFO-Newton, but its performance curves are always under those of AM, except for the final part of the HQIC scenario, where AM cannot find the optimal solution of one problem. The curves of all other methods considered are far below that of AM.

As for the quality of the returned solutions, we report in Figure 2 the cumulative distribution of the relative errors, in terms of objective value, attained by the various solvers. This plot confirms that not only AM is the fastest of the considered algorithms, but it is also, along with MIFO-Newton, the most accurate one.

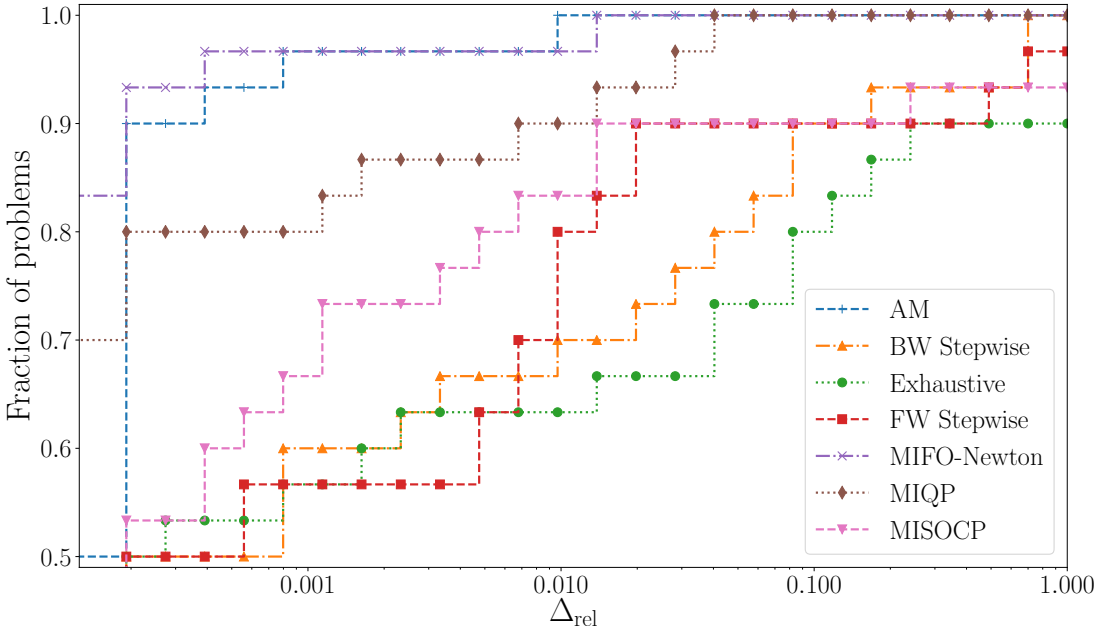


Fig. 2: Each curve represents the fraction of the 30 linear regression problems for which the corresponding solver obtains a relative error less or equal than Δ_{rel} w.r.t. the optimal value.

5.3 Autoregressive Model Selection and Fitting

Concerning the problem of auto-regressive models selection and fitting, we compared the performance of Algorithm 2 to those of commonly used techniques. In particular, we considered the strategies employed within popular automated tools such as `auto.arima()` from `R` package `forecast` [23]. This kind of tools typically provide a range of options.

First, the model section step can be carried out optimizing the approximate or the exact version of the desired GOF measure. The latter case should produce better solutions at the cost of a longer runtime. Second, in the model selection phase, all possible values of p (up to a fixed upper bound P) can be explored, or a step-wise heuristic can be adopted.

We therefore implemented the exhaustive, the forward selection and the backward elimination strategies, for selecting the order p evaluating the models through the approximate or through the exact GOF measures.

Additionally, we adapted the MISOCP model (7) to the problem of auto-regression, in the same way as Algorithm 2 is the adaptation of Algorithm 1. Altogether we thus compare AM to state-of-the-art methods and to an other method employing mixed-integer optimization.

For all algorithms employing the approximate GOF measure in the model selection phase, a refining step is carried out at the end of the process, as commonly done by state-of-the-art software. Such refining step consists of the optimization of the exact GOF measure, having fixed the order p and starting from the best solution detected in the model selection step. On the other hand, algorithms that consider the exact objective during the model selection stage don't need refinement and return the best solution found through the process.

In the implementations of AM and the MISOCP model, constraints to guarantee stationarity of solutions of order 1 and 2 were inserted. As for the management of non-stationary solutions, we did as follows. Stepwise and exhaustive

search discard altogether an order value p if the optimal solution with that order is non-stationary. In AM and MISOCP, additional constraints forbid a value of p if the mixed integer solver produces a solution of that order that is non-stationary; the process then continues in AM or restarts in MISOCP. If the final refining step, starting from a stationary solution, leads to a non-stationary solution, then the new solution is discarded and the starting point is retained as the optimal.

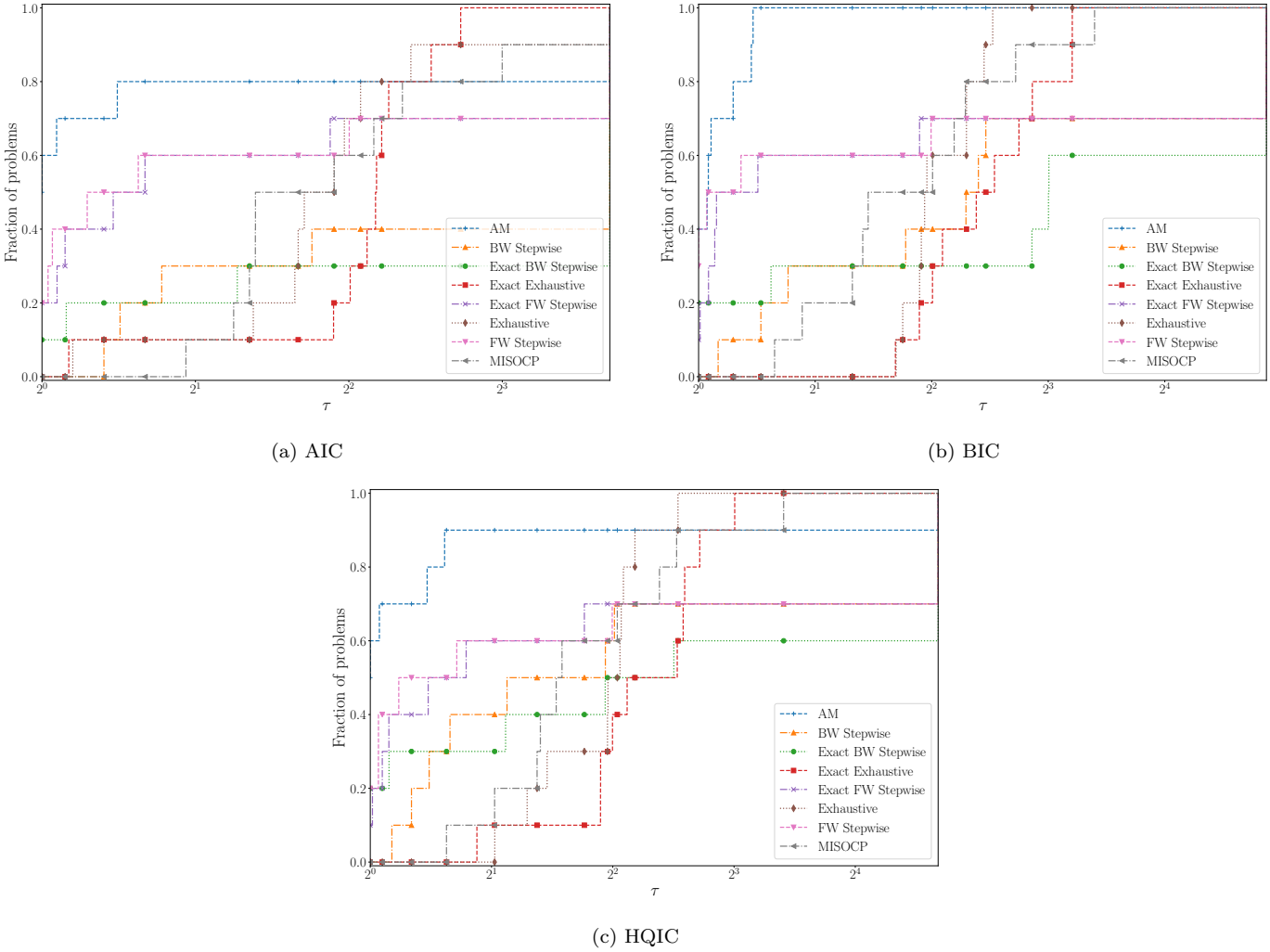
The optimization of a given GOF, be it exact or approximate, for fixed p, is carried out by means of L-BFGS-B algorithm [12]. This operation is needed by step-wise and exhaustive solvers and is also used in the final, refining phase. The value P used as an upper bound to the model order was set, for all algorithms, to 10 in problems from real time series (autoregressive models of order greater than 10 are usually not of interest in applications) and to 5 (which is a commonly employed value in software libraries, e.g. it's the default value of P in `auto.arima()`) in the case of synthetic data.

The comparison is performed over problems generated from both real and synthetic time series as described in Section 5.1. The results of the experiments are again summarized through performance profiles, shown in Figures 3 and 4. As in the previous experiments, we consider the run of an algorithm as failed if the returned solution is not optimal. We also provide in Table 4 a summary of the results of experiments on problems from real series.

Method	# of optimal solutions	total time (sec)
AM	27	75
BW Stepwise	18	134
Exact BW Stepwise	15	113
Exact Exhaustive	30	273
Exact FW Stepwise	21	121
Exhaustive	29	279
FW Stepwise	21	130
MISOCP	29	2513

Table 4: For each considered algorithm, the number of times the optimal solution was found and the sum of runtimes is reported, out of the 30 AR model fitting problems from real time series (AIC/BIC/HQIC for 10 time series). A solution is considered optimal if the relative error, in terms of objective value, is less than 10^{-3} and the order of the solution is equal to order of the optimal solution.

From Table 4, we see that our AM method is the fastest of the considered algorithms. AM doesn't always succeed at finding the optimal model, but the number of failures is very limited. We can say that it is a very good compromise between quality of the solution and speed, as the returned solution is often good and algorithms that proved to be more reliable at finding the overall optimal model require a significantly larger amount of computing time. Through a deeper analysis of the results we could observe that the error, in terms of objective value, committed by AM (when it doesn't find the optimal model) is usually small. Thus, the convenience of using AM appear even more evident.



(a) AIC

(b) BIC

(c) HQIC

Fig. 3: Performance profiles of runtimes of different algorithms on 10 AR model fitting problems from the real time series in Table 2. Three different GOF measures have been considered.

Regarding the other considered algorithms, some facts are worth to be noticed. First, the backward elimination method succeeded more often when model selection was based on the approximate GOF measure than the exact one. This result may appear puzzling. However, this can happen as the step-wise procedure, by chance, might stop too early in the approximate case and keep exploring new values of p in the exact one.

Second, we notice that, very surprisingly to us, the exact versions of enumerative algorithms outperform, in terms of runtime, the approximate versions. This behavior can in fact also be explained. The time required by L-BFGS-B to optimize the approximate GOF measure is indeed less than that needed for the exact one, but the difference is thin. On the contrary, the additional optimization performed in the refinement step of "inexact" algorithms affects significantly the total runtime.

The performance profiles in Figure 3 support the above analysis: AM is by far the most efficient method, with the only drawback that the optimal solution is not obtained in a few cases.

Now, let's turn to the experiments on synthetic time-series. Again, AM represents a solid choice, balancing speed and effectiveness. AM appears to be much faster than all other methods. In particular, all methods that attain an higher

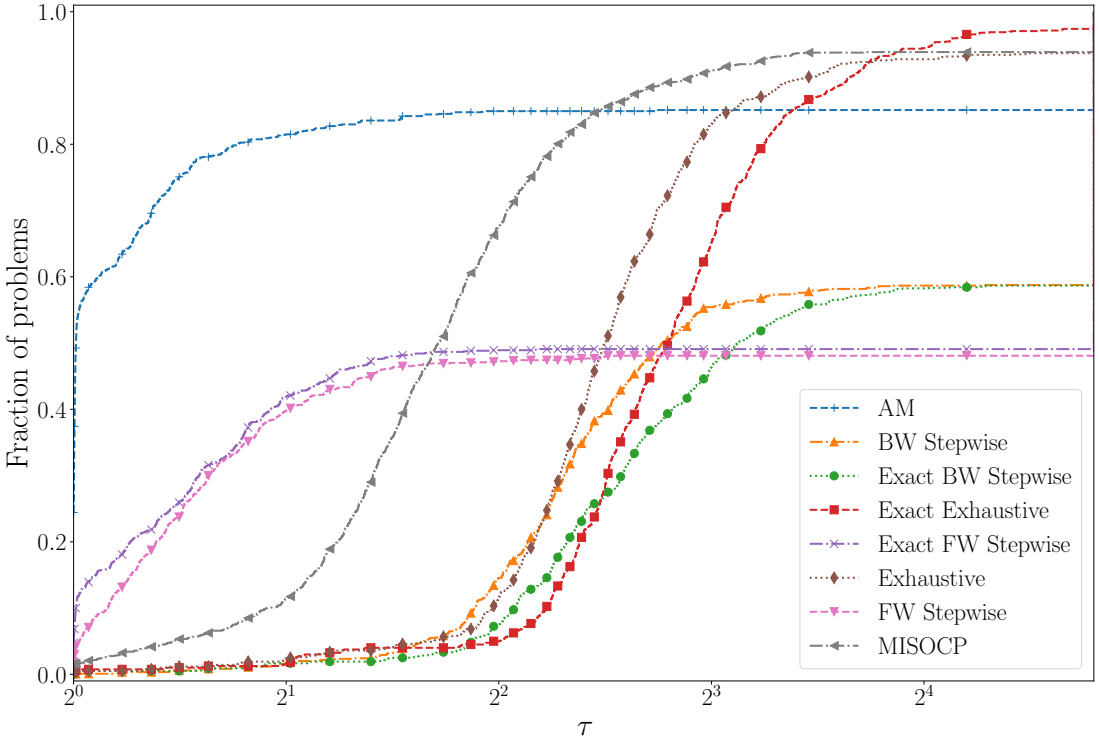


Fig. 4: Performance profiles of runtimes of different algorithms on 1200 AR model fitting problems from synthetic time series (400 series, 3 GOF measures).

success rate w.r.t. AM are significantly slower than AM. On the other hand, the forward stepwise heuristic, which is indeed quite fast, has the lowest success rate, with both exact and approximate setups.

6 Conclusions

In this paper two different, but closely related problems have been considered and a computational approach has been proposed for both of them. Both problems are composed of a linear regression component and of a discrete selection which, in one case, corresponds to finding a sparse subset of features and in the other one of a small order for an AR model. We developed, theoretically analyzed and numerically experimented a decomposition approach for both classes of problems which rely on the solution of mixed integer quadratic programming sub–problems. We have shown that our approach is capable of delivering high quality solutions in very small CPU time; we compared our approach with state of the art methods for both problems and obtained significantly improved results in most of the tests we performed. For what concerns time series models, we limited ourselves to auto-regressive ones; the extension of the proposed approach to ARMA or ARIMA models, which include a moving average component, is non trivial; research is being carried out in order to possibly extend the approach to these models.

Acknowledgments

We are deeply indebted to the Associate Editor for having pointed out an important reference during the first review of this paper. This suggestion led us to totally revise and significantly expand the scope of this paper. We are also grateful to the reviewers for their useful and constructive comments.

References

1. H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
2. H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.
3. D. A. Anderson, K. P. Burnham, D. Anderson, and K. P. Burnham. Model selection and inference: a practical information-theoretic approach, 1998.
4. A. Bagirov, C. Clausen, and M. Kohler. An algorithm for the estimation of a regression function by continuous piecewise linear functions. *Computational Optimization and Applications*, 45(1):159–179, Jan 2010.
5. D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2016.
6. D. Bertsimas and A. King. Or forum—an algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2015.
7. D. Bertsimas, A. King, R. Mazumder, et al. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.
8. D. Bertsimas and R. Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22, 2009.
9. D. Bertsimas and B. Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *arXiv preprint arXiv:1709.10029*, 2017.
10. G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
11. H. Bozdogan. Akaike's information criterion and recent developments in information complexity. *Journal of mathematical psychology*, 44(1):62–91, 2000.

12. R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.
13. A. Cozad, N. V. Sahinidis, and D. C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.
14. A. M. De Livera, R. J. Hyndman, and R. D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.
15. E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.
16. D. Dua and C. Graff. UCI machine learning repository. 2017. University of California, Irvine, School of Information and Computer Sciences. `http://archive.ics.uci.edu/ml`.
17. M. Efroymson. Multiple regression analysis. *Mathematical methods for digital computers*, pages 191–203, 1960.
18. M. Garside. Some computational procedures for the best subset problem. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 20(1):8–15, 1971.
19. A. Gómez and O. Prokopyev. A mixed-integer fractional optimization approach to best subset selection. Technical Report Optimization On Line, 6795, Swanson School of Engineering, University of Pittsburgh, 2018.
20. Gurobi Optimization LLC. Gurobi optimizer reference manual, 2018. `http://www.gurobi.com`.
21. J. D. Hamilton. *Time series analysis*, volume 2. Princeton University Press, Princeton, NJ, 1994.
22. E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):190–195, 1979.
23. R. J. Hyndman, Y. Khandakar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics, 2007.
24. K. Kimura and H. Waki. Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program. *Optimization Methods and Software*, 33(3):633–649, 2018.
25. S. Konishi and G. Kitagawa. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.
26. D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.
27. M. C. Medeiros, M. G. Resende, and A. Veiga. Piecewise linear time series estimation with GRASP. *Computational Optimization and Applications*, 19(2):127–144, Jul 2001.
28. A. Miller. *Subset selection in regression*. Chapman and Hall/CRC, 2002.
29. R. Miyashiro and Y. Takano. Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247(3):721–731, 2015.
30. R. Miyashiro and Y. Takano. Subset selection by Mallows' Cp: A mixed integer programming approach. *Expert Systems with Applications*, 42(1):325–331, 2015.
31. B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.
32. D. R. Osborn, A. P. Chui, J. P. Smith, and C. R. Birchenhall. Seasonality and the order of integration for consumption. *Oxford Bulletin of Economics and Statistics*, 50(4):361–377, 1988.
33. T. Sato, Y. Takano, R. Miyashiro, and A. Yoshise. Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, 64:865–880, 2016.
34. G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.
35. X. Shen, W. Pan, Y. Zhu, and H. Zhou. On constrained and regularized high-dimensional regression. *Annals of the Institute of Statistical Mathematics*, 65(5):807–832, 2013.
36. R. Shibata. Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*, 63(1):117–126, 1976.

37. R. C. Team et al. R: A language and environment for statistical computing. 2013.

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

39. Z. T. Wilson and N. V. Sahinidis. The ALAMO approach to machine learning. *Computers & Chemical Engineering*, 106:785–795, 2017.

40. Z. Zheng, Y. Fan, and J. Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014.