lmSubsets: Efficient Computation of Variable-Subsets Linear Regression in R

Marc Hofmann
TODO: affiliation

Cristian Gatu
TODO: affiliation

Erricos J. Kontoghiorghes
Birbeck College

Ana Colubi
University of Oviedo

Achim Zeileis
Universität Innsbruck

Abstract

TODO: write abstract

Keywords: linear regression, model selection, variable selection, best subset regression, R.

1. Introduction

TODO: Some comments and references to applications might be useful here.

An important problem in statistical modeling is that of subset selection regression or, equivalently, of finding the best regression equation (Hastie *et al.* 2001). Given a set of possible variables to be included in the regression, the problem consists in selecting a subset that optimizes some statistical criterion. The latter originates in the estimation of the corresponding submodel (Miller 2002). Consider the standard regression model

$$y = X\beta + \epsilon, \tag{1}$$

where $y \in \mathbb{R}^M$ is the output variable, $X \in \mathbb{R}^{M \times N}$ is the regressor matrix of full column rank, $\beta \in \mathbb{R}^N$ is the coefficient vector, and $\epsilon \in \mathbb{R}^M$ is the noise vector. The ordinary least squares (OLS) estimator of β is the solution of

$$\hat{\beta}_{\text{OLS}} = \underset{\beta}{\operatorname{argmin}} \operatorname{RSS}(\beta), \tag{2}$$

where the residual sum of squares (RSS) of β is given by

$$\operatorname{RSS}(\beta) = \|y - X\beta\|_2^2. \tag{3}$$

That is, $\hat{\beta}_{\text{OLS}}$ minimizes the norm of the residual vector. The computation of the RSS does not require to explicitly determine β; the former can be done by means of orthogonal matrix decomposition methods and is numerically stable (Golub and Van Loan 1996).

Let $V = \{1, \ldots, N\}$ denote the set of all possible variables. A subset model (or submodel) is denoted by S, $S \subseteq V$. Given a criterion function f, the *best-subset selection problem* consists in solving

$$S^* = \underset{S \subseteq V}{\operatorname{argmin}} f(S). \tag{4}$$

Here, the value $f(S) = F(n, \rho)$ is seen as a function of $n = |S|$ and $\rho = \text{RSS}(S)$, the number of selected variables and the RSS of the OLS estimator of S, respectively. Furthermore, it is assumed that $f(S)$ is monotonic with respect to $\text{RSS}(S)$ for fixed n, that is

$$\text{RSS}(S_1) \leq \text{RSS}(S_2) \Rightarrow f(S_1) \leq f(S_2), \quad |S_1| = |S_2|. \tag{5}$$

Common selection criteria exhibit this property, such as those belonging to the AIC family defined by the formula

$$\text{AIC}_k = M + M \log 2\pi + M \log(\text{RSS}/M) + k(n+1), \tag{6}$$

where the scalar k represents a *penalty per parameter* ($k > 0$). The usual AIC and BIC are obtained for $k = 2$ and $k = \log M$, respectively (Miller 2002). It follows that (4) is equivalent to

$$S^* = S^*_\nu, \quad \text{where} \quad \nu = \underset{n}{\text{argmin}}\, f(S^*_n)$$

and

$$S^*_n = \underset{|S|=n}{\text{argmin}}\, \text{RSS}(S) \quad \text{for} \quad n = 1, \ldots, N. \tag{7}$$

Finding the solution to (7) is called the *all-subsets selection problem*. Thus, solving (4) can be seen as an indirect, two-stage procedure:

Stage 1 For each subset size n, find the subset S^*_n ($|S^*_n| = n$) with the smallest RSS.

Stage 2 Compute $f(S^*_n)$ for all n, and determine ν such that $f(S^*_\nu)$ is minimal.

Note that the computational strategy may be optimized for a specific selection criterion when solving the best-subset selection problem (4) directly, thus lowering the compuational cost. On the other hand, by explicitly solving the all-subsets regression problem (7) once and for all (Stage 1), the list of all N submodels is made readily available for further exploration: the evaluation of multiple criterion functions (e.g., AIC and BIC), or conducting a more elaborate statistical inference, can be done at a negligible cost (Stage 2). Thus, it can be advisable to adopt a two-stage approach in the case of a broader and more thorough statistical investigation.

Brute-force (or exhaustive) search procedures that enumerate all possible subsets become intractable even for a modest number of variables. Exact algorithms must employ techniques to reduce the size of the search space – i.e., the number of enumerated subsets – in order to tackle larger problems. Heuristic algorithms renounce optimality in order to decrease execution times: they are designed for solving a problem more quickly, but make no guarantees on the quality of the solution produced; genetic algorithms and simulated annealing count among the well-known heuristic algorithms. Approximation algorithms, on the other hand, return a solution that provably lies within well specified bounds of the optimum.

Several packages that deal with variable subset selection are available on the R platform. The package **leaps** (Lumley and Miller 2009) implements exact, non-exhaustive algorithms for subset regression based on Miller (2002). Exhaustive algorithms have been considered within the context of generalized linear models (package **bestglm**, McLeod and Xu 2014). The package **subselect** proposes simulated annealing algorithms based on the work of Duarte Silva (2001). Furthermore, genetic algorithms for generalized linear models and beyond have

been implemented by Calcagno and de Mazancourt (2010, package **glmulti**) and Wolters (2015, package **kofnGA**). Non-exact algorithms for regularized estimation of parametric models with automatic variable selection performed by lasso or elastic net estimation for generalized linear models have been investigated by Friedman *et al.* (2010).

Here, the **lmSubsets** package (Hofmann *et al.* 2016) for exact variable-subset regression is presented. It offers methods for solving both the *best-subset* (4) and the *all-subsets* (7) selection problems. It implements the algorithms presented by Gatu and Kontoghiorghes (2006) and Hofmann *et al.* (2007). A branch-and-bound strategy is employed to reduce the size of the search space. The package further proposes approximation methods that compute non-exact solutions very quickly while giving guarantees on the quality of the result. The core of the package is written in C++. The package is available for the R system for statistical computing (R Core Team 2016) from The Comprehensive R Archive Network at `https://CRAN.R-project.org/package=lmSubsets`.

Section 2 reviews the theoretical background and the underlying algorithms. The package's R interface is presented in Section 3. A usage example is given in Section 4, while benchmark results are illustrated in Section 6.

`TODO: No discussion section?`

2. Computational strategies

The linear regression model (1) has 2^N possible subset models which can be efficiently organized in a regression tree. A dropping column algorithm (DCA) was devised as a straightforward approach to solve the all-subsets selection problem (7). The DCA evaluates all possible variable subsets by traversing a regression tree consisting of 2^{N-1} nodes (Gatu and Kontoghiorghes 2003; Smith and Bremner 1989).

Each node (S, k) of the tree corresponds to a subset $S = \{s_1, \ldots, s_n\}$ of n variables and an index k ($k = 0, \ldots, n-1$). The root node $(V, 0)$ corresponds to the full set of variables. For every visited node (S, k), the RSS of the $n - k$ subleading models corresponding to the subsets $\{s_1, \ldots, s_{k+1}\}, \ldots, \{s_1, \ldots, s_n\}$ are reported. Child nodes are generated by dropping (deleting) a variable:

$$\mathsf{drop}(S, j) = (S \setminus \{s_j\}, j-1), \quad j = k+1, \ldots, n-1.$$

Numerically, this is equivalent to downdating an orthogonal matrix decomposition after a column has been deleted (Golub and Van Loan 1996; Kontoghiorghes 2000; Smith and Bremner 1989). Givens rotations are employed to efficiently move from one node to another. The DCA maintains a subset table r with N entries; entry r_n contains the RSS and the variable subset of the current-best submodel of size n (Gatu and Kontoghiorghes 2006; Hofmann *et al.* 2007). Figure 1 illustrates a regression tree for $N = 5$ variables. The index k is symbolized by a bullet (•). The subleading models are listed in each node.

The DCA is computationally demanding, with a theoretical time complexity of $O(2^N)$. A branch-and-bound algorithm (BBA) has been devised to reduce the number of generated nodes by cutting subtrees which do not contribute to the current-best solution. It relies on the fundamental property that the RSS increases when variables are deleted from a regression model, that is:

$$S_1 \subseteq S_2 \Rightarrow \mathrm{RSS}(S_1) \geq \mathrm{RSS}(S_2).$$

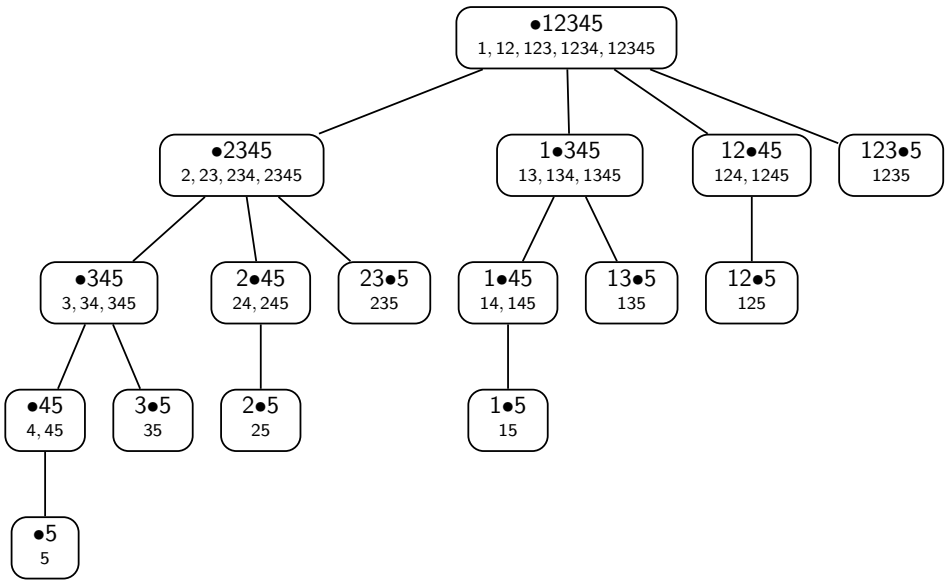

Figure 1: All-subsets regression tree (with subleading models) for $N = 5$ variables.

A cutting test is employed to determine which parts of the DCA tree are redundant: A new node $\mathsf{drop}(S, j)$ is generated only if $\mathrm{RSS}(S) < r_j$ $(j = k+1, \ldots, n-1)$. The quantity $\mathrm{RSS}(S)$ is called the *bound* of the subtree rooted in (S, k): no subset model extracted from the subtree can have a smaller RSS (Gatu and Kontoghiorghes 2006). Note that the BBA is an exact algorithm, i.e., it computes the optimal solution of the all-subsets regression problem (7).

To further reduce the computational cost, the all-subsets regression problem can be restricted to a range of submodel sizes (Hofmann *et al.* 2007). In this case, the problem (7) is reformulated as

$$S_n^* = \underset{|S|=n}{\operatorname{argmin}} \mathrm{RSS}(S) \quad \text{for} \quad n = n_{\min}, \ldots, n_{\max}, \tag{8}$$

where $n_{\min}$ and $n_{\max}$ are the *subrange limits* $(1 \leq n_{\min} \leq n_{\max} \leq N)$. The search will span only a part of the DCA regression tree. Specifically, nodes (S, k) are not computed if $|S| < n_{\min}$ or $k \geq n_{\max}$.

The size of subtrees rooted in the same level decreases exponentially from left to right. In order to encourage the pruning of large subtrees by the BBA cutting test, the variables in a given node can be ordered such that a child node will always have a larger RSS (i.e., bound) than its right siblings (Gatu and Kontoghiorghes 2006). This strategy can be applied in nodes of arbitrary depth. However, computing the variable bounds incurs a computational overhead. Thus, it is not advisable to indiscriminately preorder variables. A parameter – the preordering radius p – has been introduced to control the degree of preordering (Hofmann *et al.* 2007). It accepts a value between $p = 0$ (no preordering) and $p = N$ (preordering in all nodes); when $p = 1$, preordering is performed in the root node only. Typically, $p = \lfloor N/3 \rfloor$ produces good results.

The computational efficiency of the BBA is improved by allowing the algorithm to prune non-redundant portions of the regression tree. The so-called heuristic branch-and-bound algorithm (HBBA) relaxes the cutting test by employing a set of tolerance parameters $\tau_n \geq 0$

$(n = 1, \ldots, N)$, one for every submodel size. A node $\mathsf{drop}(S, j)$ is generated only if there exists at least one i such that

$$(1 + \tau_i) \cdot \mathrm{RSS}(S) < r_i, \quad i = j, \ldots, n - 1. \tag{9}$$

The algorithm is non-exact if $\tau_n > 0$ for any n, meaning that the computed solution is not guaranteed to be optimal. The algorithm cuts subtrees the more aggressively the greater the value of τ_n, thus increasing the computational efficiency. The advantage of the HBBA over heuristic algorithms is that the relative error of the solution is bounded by the tolerance parameter (Gatu and Kontoghiorghes 2006; Hofmann *et al.* 2007), thus giving the user control over the tradeoff between solution quality and speed of execution. Note that "heuristic BBA" is a misnomer, as the algorithm does not belong to the class of heuristic algorithms at all; rather, it is an approximation algorithm.

The DCA and its derivatives report the N subset models with the lowest RSS, one for each subset size. The user can then analyze the list of returned subsets to determine the "best" subset, for example by evaluating some criterion function. This approach is practical but not necessarily the most efficient. The f-BBA specializes the cutting test of the standard BBA for a particular criterion function $f(S) = F(n, \rho)$, where $n = |S|$ and $\rho = \mathrm{RSS}(S)$, under the condition that it satisfies the monotonicity property (5). Specifically, a node $\mathsf{drop}(S, j)$ is generated if and only if

$$F(j, \mathrm{RSS}(S)) < r_f,$$

where r_f is the *single* current-best solution. This results in a more "informed" cutting test, and in a smaller number of generated nodes.

3. Implementation in R

The R package **lmSubsets** provides a simple framework for variable subset selection in linear regression. Two S3 classes are defined, namely "`lmSubsets`" and "`lmSelect`", that address all-subsets (7) and best-subset (4) selection, respectively. The package offers R's standard formula interface: linear models can be specified by means of a symbolic formula, and possibly a data frame. The model specification is resolved into a regressor matrix and a response vector, which are forwarded to low-level workhorse functions for actual processing, together with optional arguments that serve to further specify the selection problem. Routines to extract the best subsets from an all-subsets regression solution (i.e., to convert an "`lmSubsets`" to an "`lmSelect`" object) are also provided. An overview of the package structure is given in Table 1.

3.1. Specifying the selection problem

The default methods are closely modeled after R's standard `lm` function: they can be called with any entity that can be coerced to a `formula` object (Chambers and Hastie 1992). The `formula` object declares the dependent and independent variables, which are typically taken from a `data.frame` specified by the user. For example, the call

```
lmSubsets(mortality ~ precipitation + temperature1 + temperature7 +
  age + household + education + housing + population + noncauc +
```

S3 class	Methods and functions	Description
"lmSubsets"	lmSubsets()	All-subsets selection (generic function)
	lmSubsets.default()	Standard formula interface
	lmSubsets_fit()	Workhorse function
"lmSelect"	lmSelect()	Best-subset selection (generic function)
	lmSelect.default()	Standard formula interface
	lmSelect.lmSubsets()	Conversion method
	lmSelect_fit()	Workhorse function
	lmSubsets_select()	Explicit conversion function

Table 1: Package structure.

```
whitecollar + income + hydrocarbon + nox + so2 + humidity,
data = AirPollution)
```

specifies a response variable (`mortality`) and fifteen predictor variables, all taken from the `AirPollution` dataset (Miller 2002). It is common to shorten the call by emloying R's practical "dot-notation":

```
lmSubsets(mortality ~ ., data = AirPollution),
```

where the dot (`.`) stands for "all variables not mentioned in the left-hand side of the formula". By default, an intercept term is included in the model; that is, the call in the previous example is equivalent to

```
lmSubsets(mortality ~ . + 1, data = AirPollution).
```

To discard the intercept, the call may be rewritten as follows:

```
lmSubsets(mortality ~ . - 1, data = AirPollution).
```

Submodels can be rejected based on the presence or absence of certain independent variables. The parameter `include` specifies that all submodels must contain one or several variables. In the following example, only submodels containing the variable `noncauc` are retained:

```
lmSubsets(mortality ~ ., include = "noncauc", data = AirPollution).
```

Conversely, the `exclude` parameter can be employed to discard a specific set of variables, as in the following example:

```
lmSubsets(mortality ~ ., exclude = "whitecollar", data = AirPollution).
```

The same effect can be achieved by rewriting the formula as follows:

```
lmSubsets(mortality ~ . - whitecollar, data = AirPollution).
```

The `include` and `exclude` parameters may be used in combination, and both may specify more than one variable (e.g., `include = c("noncauc", "whitecollar")`).

The criterion used for best-subset selection is evaluated following the expression

$$-2 \cdot \texttt{logLik} + \texttt{penalty} \cdot \texttt{npar},$$

where `penalty` is the *penalty per model parameter* defined in (6), `logLik` the log-likelihood of the fitted model, and `npar` the number of model parameters (including the error variance). The `penalty` value indicates how strongly model parameters are penalized, with large values favoring parsimonious models. When `penalty` $= 2$, the criterion corresponds to Akaike's information criterion (AIC, Akaike 1974); when `penalty` $= \log($`nobs`$)$, to Schwarz's Bayesian information criterion (BIC, Schwarz 1978), where `nobs` is the number of observations. For example, either one of

```
lmSelect(mortality ~ ., data = AirPollution, penalty = 2)
```

and

```
lmSelect(mortality ~ ., data = AirPollution, penalty = "AIC")
```

will select the best submodel according to the usual AIC; by default, `lmSelect()` employs the BIC.

3.2. Core functions

The high-level interface methods process the model specification before dispatching the call to one of two low-level core functions, passing along a regressor matrix `x` and a response vector `y`, together with problem-specific arguments. The core functions act as wrappers around the C++ library, and are declared as

```
lmSubsets_fit(x, y, weights = NULL, offset = NULL, include = NULL,
  exclude = NULL, nmin = NULL, nmax = NULL, tolerance = 0, pradius = NULL,
  nbest = 1, ..., .algo = "phbba")
```

and

```
lmSelect_fit(x, y, weights = NULL, offset = NULL, include = NULL,
  exclude = NULL, penalty = "BIC", tolerance = 0, pradius = NULL,
  nbest = 1, ..., .algo = "phbba").
```

The parameters are summarized in Table 2.

The `weights` and `offset` parameters correspond to the homonymous parameters of the `lm` function. The `include` and `exclude` parameters allow the user to specify variables that are to be included in, or excluded from all candidate models. They are either logical vectors – with each entry corresponding to one variable – or are automatically expanded if given in the form of an integer vector (i.e., set of variable indices) or character vector (i.e., set of variable names).

For a large number of variables (see Section 6), execution times may become intractable. In order to speed up the execution, either the search space can be reduced, or one may settle

Parameter	Description	Canonical representation	
`x`	Data matrix	`numeric[nobs,nvar]`	
`y`	Response variable	`numeric[nobs]`	
`weights`	Model weights	`numeric[nobs]`	
`offset`	Model offset	`numeric[nvar]`	
`include`	Regressors to force in	`logical[nvar]`	
`exclude`	Regressors to force out	`logical[nvar]`	
`nmin`	Min. number of regressors	`integer[1]`	for `lmSubsets()` only
`nmax`	Max. number of regressors	`integer[1]`	for `lmSubsets()` only
`tolerance`	BBA tolerance parameters	`numeric[nvar]`	for `lmSubsets()`
		`numeric[1]`	for `lmSelect()`
`pradius`	Preordering radius	`integer[1]`	
`nbest`	Number of best subsets	`integer[1]`	
`.algo`	Algorithm to execute	`character[1]`	

Table 2: Core parameters.

for a non-exact solution. In the first approach, the user may specify values for the `nmin` and `nmax` parameters as defined in (8), in which case submodels with less than `nmin` or more than `nmax` variables are discarded. Well-defined regions of the regression tree can be ignored by the selection algorithm, and the search space is thus reduced.

In the second approach, expectations with respect to the solution quality are lowered, i.e., non-optimal solutions are *tolerated*. This is indicated by passing a numeric value – typically between 0 and 1 – to the `tolerance` parameter, which will be used by the HBBA cutting test (9) to prune the search tree more aggressively. The solution produced by the HBBA satisfies the following relationship:

$$f(S) \leq (1 + \texttt{tolerance}) \cdot f(S^*),$$

where S is the returned solution, S^* the optimal (theoretical) solution, and f the value of a submodel (e.g., deviance, AIC). The `lmSubsets_fit()` function accepts a vector of tolerances, with one entry for each subset size.

The `nbest` parameter controls how many submodels (per subset size) are retained. In the case of `lmSubsets_fit()`, a two-dimensional result set is constructed with `nbest` submodels for each subset size, while in the case of `lmSelect_fit()`, a one-dimensional sequence of `nbest` submodels is handed back to the user.

The `pradius` parameter serves to specify the desired preordering radius. The algorithm employs a default value of $\lfloor \texttt{nvar}/3 \rfloor$. The need to set this parameter directly should rarely arise; please refer to Section 2 for further information. The `.algo` parameter serves to specify the computational algorithm to be employed. This parameter is used for testing purposes only and should never be set by the user.

3.3. Extracting submodels

The user is handed back a result object that encapsulates the solution to an all-subsets (class "`lmSubsets`") or best-subset (class "`lmSelect`") selection problem. An object of class

Component	Description	Canonical representation	
`nobs`	Number of observations	`integer[1]`	
`nvar`	Number of regressors	`integer[1]`	
`weights`	Weights used	`numeric[nobs]`	
`offset`	Offset used	`numeric[nobs]`	
`intercept`	Intercept flag	`logical[1]`	
`include`	Regressors forced in	`logical[nvar]`	
`exclude`	Regressors forced out	`logical[nvar]`	
`penalty`	Penalty used	`numeric[nvar]`	for "`lmSelect`" only
`tolerance`	Tolerances used	`numeric[nvar]`	
`nbest`	Number of best subsets	`integer[1]`	
`df`	Degrees of freedom	`integer[nbest,nvar]`	for "`lmSubsets`"
		`integer[nbest]`	for "`lmSelect`"
`rss`	Residual sum of squares	`numeric[nbest,nvar]`	for "`lmSubsets`"
		`numeric[nbest]`	for "`lmSelect`"
`which`	Selected regressors	`logical[nvar,nbest,nvar]`	for "`lmSubsets`"
		`logical[nvar,nbest]`	for "`lmSelect`"

Table 3: Components of "`lmSubsets`" and "`lmSelect`" objects.

"`lmSubsets`" represents a two-dimensional `nbest` × `nvar` set of submodels; an object of class "`lmSelect`", a linear sequence of `nbest` submodels. Problem-specific information is stored alongside the selected submodels. Table 3 summarizes the components of the result objects.

A wide range of standard methods to visualize, summarize, and extract information are provided (see Table 4). The `print()`, `plot()`, and `summary()` methods give the user a compact overview – either textual or graphical – of the information gathered on the selected submodels in order to help him identify "good" candidates. The remaining extractor functions behave in the usual way, and can be used to extract variable names, coefficients, covariance matrices, fitted values, etc.

In order to designate a submodel, "`lmSubsets`" methods provide the parameters `size` and `best` to specify the number of regressors in and the ranking of the desired submodel, respectively. The user must always indicate the desired `size`, while `best` defaults to `1` if left unspecified. Moreover, the name of a selection criterion (`"AIC"` or `"BIC"`) may be given as the `size` argument, in which case the submodel with the smallest criterion value is extracted. For "`lmSelect`" methods, the `size` parameter has no meaning and is not defined. Lastly, methods that return scalar values – i.e., `deviance()`, `logLik()`, `AIC()`, `BIC()` – can extract more than one submodel at a time if passed a numeric vector as an argument to either `size` (e.g., `size = 5:10`) or `best` (e.g., `best = 1:3`).

4. Case study: Variable selection for weather forecasting

Over the last decades the field of weather forecasting made steady and substantial improvements especially through improvements in numerical weather prediction (NWP) models (Bauer *et al.* 2015). Starting from Glahn and Lowry (1972) the outputs from these physically-based large-scale (typically global) NWP models is statistically post-processed to

Method	Description	
`print()`	Print object	
`plot()`	Plot RSS (and information criteria)	
`image()`	Heatmap of selected regressors	for "`lmSubsets`" only
`summary()`	Summary statistics	
`variable.names()`	Extract variables names	
`formula()`	Extract formula object	
`model.frame()`	Extract (full) model frame	
`model.matrix()`	Extract model matrix	
`model.response()`	Extract model response	
`refit()`	Fit subset "`lm`" model	
`coef()`	Extract regression coefficients	
`vcov()`	Extract covariance matrix	
`fitted()`	Extract fitted values	
`residuals()`	Extract residual values	
`deviance()`	Extract deviance (RSS)	
`logLik()`	Extract log-likelihood	
`AIC()`	Extract AIC values	
`BIC()`	Extract BIC values	

Table 4: S3 methods for "`lmSubsets`" and "`lmSelect`" objects.

correct small-scale biases and obtain predictions for specific locations. Below we use such model output statistics (MOS) to predict temperature at a specific station (Innsbruck Airport, Austria) based on a wide range of NWP quantities from the nearest NWP grid point. Variable subset selection is relevant here because it is not obvious which quantities beyond the temperature NWP forecasts should enter the MOS regression.

More specifically, we model 00UTC temperature observations (in degree Celsius) based on the corresponding 24-hour reforecast ensemble means from the Global Ensemble Forecast System (GEFS, Hamill *et al.* 2013) for meteorological station Innsbruck Airport (11120; 47.260, 11.357) from 2011-01-01 to 2015-12-31. The data frame `IbkTemperature` contains 1824 daily observations/forecasts for the observed response, 36 NWP outputs, and five deterministic time trend/season patterns that are available as potential regressors. The NWP variables include several temperature quantities (in degree Kelvin, e.g., 2-meter, minimum, maximum, soil) as well as several quantities capturing precipitation, wind, and fluxes among others. See `?IbkTemperature` for more details. The data from the NOAA (United States National Oceanic and Atmospheric Administration) are obtained from `http://www.esrl.noaa.gov/psd/forecasts/reforecast2/` (reforecasts) and `http://www.ogimet.com/synops.phtml.en` (observations), respectively.

To start the analysis the data from the **lmSubsets** package can be loaded and for simplicity a couple of days with some missing values are omitted.

```
R> data("IbkTemperature", package = "lmSubsets")
R> IbkTemperature <- na.omit(IbkTemperature)
```

First, a simple climatological model for the temperature (`temp`) with a linear trend (`time`)

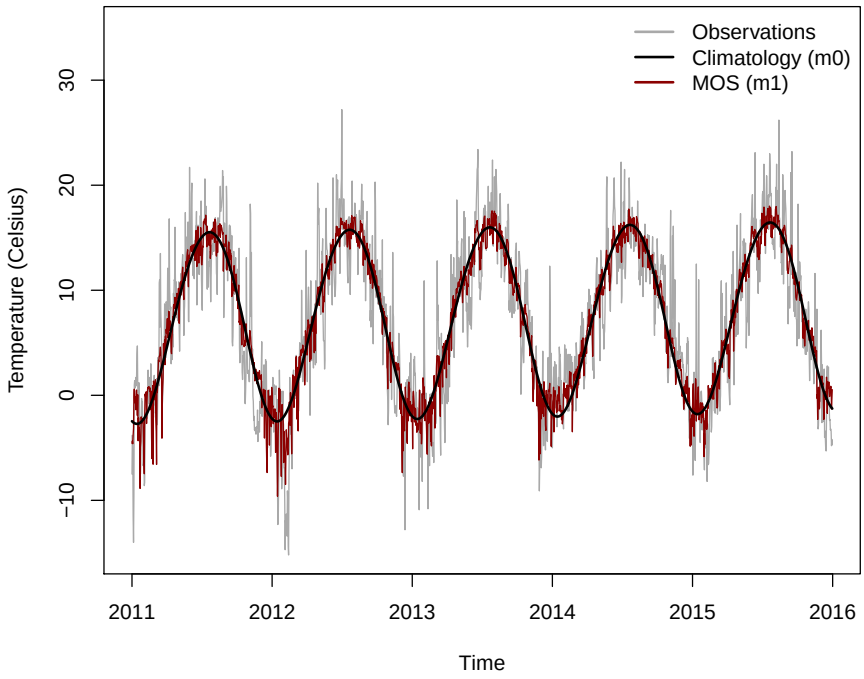

Figure 2: Observed temperature at Innsbruck Airport (gray) and fitted values from the climatological model (`m0`, black) and the simple MOS (`m1`, red).

and a harmonic seasonal pattern (`sin`/`cos` for the annual and `sin2`/`cos2` for the bi-annual frequencies).

```
R> m0 <- lm(temp ~ time + sin + cos + sin2 + cos2, data = IbkTemperature)
```

This model does not make use of any NWP outputs and is used as a basic reference model against which the subsequent MOS models can be compared. Second, the model is updated to a simple MOS by including the most obvious direct model output – 2-meter temperature (`t2m`) – in addition to the season/trend regressors.

```
R> m1 <- update(m0, . ~ . + t2m)
```

A graphical comparison of the raw data and the fitted values from both models is provided in Figure 2. The corresponding estimated coefficients (and standard errors) are shown in Table 5 (produced with **memisc**, Elff 2016). This shows that, not surprisingly, inclusion of the 2-meter temperature leads to substantial (and highly significant) improvements. The season/trend coefficients are dampened but remain significant which means that not all seasonal temperature pattern at Innsbruck Airport are resolved in the coarse NWP grid.

Subsequently, we now try to improve the MOS by not only including the direct model output for 2-meter temperature but also further NWP model outputs. As a starting point we force the regressors from `m1` into the model and use all-subsets regression to select the relevant regressors from the remaining 35 NWP variables.

```
R> ms2 <- lmSubsets(temp ~ ., data = IbkTemperature,
```

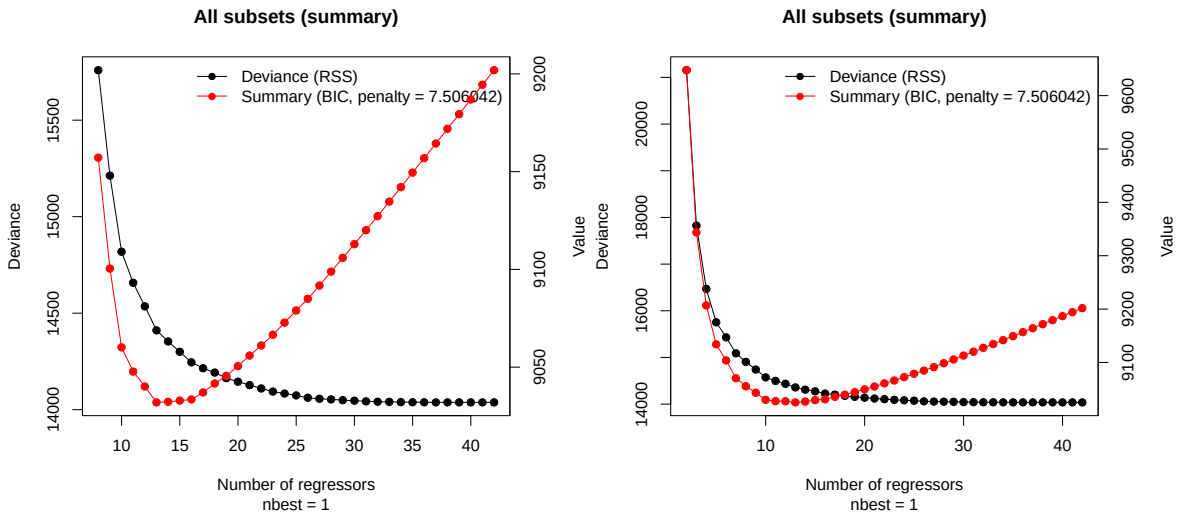

Figure 3: Best RSS and associated BIC for all subset sizes considered in `ms2` (left) and `ms3` (right).

```
+    include = c("t2m", "time", "sin", "cos", "sin2", "cos2"))
R> m2 <- refit(ms2, size = "BIC")
```

After obtaining the all-subsets regression `ms2` with `lmSubsets()` the best BIC solution is extracted and turned into a "`lm`" model with `refit()`. The more costly all-subsets regression is solved here to gain more insights into the selected variables not only for the best-BIC solution but also other models.

To assess whether our initial MOS strategy in `m1` (and forced into `m2`) is really the most suitable we also carry out another all-subsets regression without restricting the search space.

```
R> ms3 <- lmSubsets(temp ~ ., data = IbkTemperature)
R> m3 <- refit(ms3, size = "BIC")
```

Obtaining `ms3` is computationally somewhat more costly than `ms2` but still very fast taking only a couple of seconds on standard PCs.

To assess the subset selections, Figure 3 shows a graphical summary of best RSS and associated BIC for all subset sizes considered in `ms2` and `ms3`, respectively. The plots can be easily produced by `plot(summary(...))` and start with subset size 8 for `ms2` because seven variables are forced into the model while in `ms3` only the intercept is always included. For both models the RSS and BIC curves look rather similar and the best-BIC models both have 13 regressors.

The corresponding selected variables can be seen in Table 5 produced by `mtable(m1, m2, m3, m4)`. This shows that both `m2` and `m3` are rather similar with respect to the selected variables and corresponding coefficients. However, interestingly the direct model output `t2m` is not selected for `m3` and instead the soil temperature (`st`) as well as the maximal 2-meter temperature (`tmax2m`) and temperature on the so-called 2 PVU surface (`t2pvu`) are used which are selected in addition to `t2m` in `m2`. Additionally, various other meteorological quantities are selected that improve the forecasting model further, e.g., soil moisture `vsmc`) among others.

	m0	m1	m2	m3
(Intercept)	−458.732***	−345.252**	−666.584***	−661.700***
	(119.936)	(109.212)	(95.349)	(95.225)
time	0.231***	0.132*	0.149**	0.147**
	(0.060)	(0.054)	(0.047)	(0.047)
sin	−2.433***	−1.234***	0.522***	0.811***
	(0.121)	(0.126)	(0.147)	(0.120)
cos	−8.716***	−6.329***	−0.812**	
	(0.120)	(0.164)	(0.273)	
sin2	0.051	0.240*	−0.794***	−0.870***
	(0.120)	(0.110)	(0.119)	(0.118)
cos2	−0.380**	−0.332**	−1.067***	−1.128***
	(0.120)	(0.109)	(0.101)	(0.097)
t2m		0.318***	0.055	
		(0.016)	(0.029)	
sshnf			0.016***	0.018***
			(0.004)	(0.004)
vsmc			20.200***	20.181***
			(3.115)	(3.106)
tmax2m			0.145***	0.181***
			(0.037)	(0.023)
st			1.077***	1.142***
			(0.051)	(0.043)
wr			0.450***	0.505***
			(0.109)	(0.103)
t2pvu			0.064***	0.149***
			(0.011)	(0.028)
mslp				−0.000***
				(0.000)
p2pvu				−0.000**
				(0.000)
AIC	9838.5	9493.6	8954.9	8948.2
BIC	9877.1	9537.7	9032.0	9025.3
Deviance	23605.3	19506.5	14411.1	14357.9
sigma	3.6	3.3	2.8	2.8
R-squared	0.8	0.8	0.9	0.9

Table 5: Estimated regression coefficients (and standard errors) along with further summary statistics for the climatological model `m0` and the three MOS models (`m1`–`m3`).

To gain further insight into the best-subset selection for various sizes the `image()` method is useful. Figure 4 and Figure 5 show the results for `ms2` and `ms3`, respectively, for subset sizes up to 20 variables. The dark cells in the heatmap show the variables that are selected while the best-BIC solution is highlighted with a red rectangle and by underlining the variable names in the x-axis labels. While there are many similarities between the patterns shown,

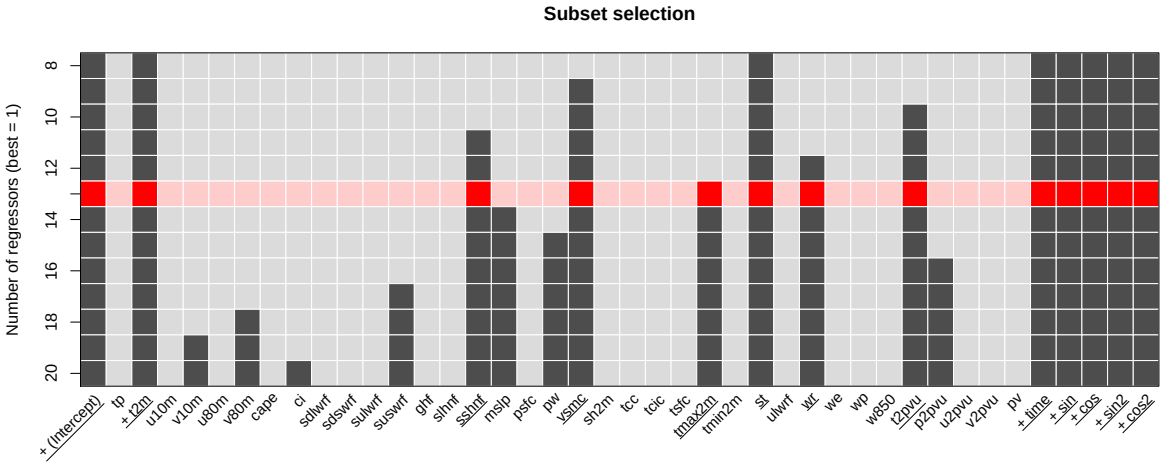

Figure 4: Subset selection for `ms2`.

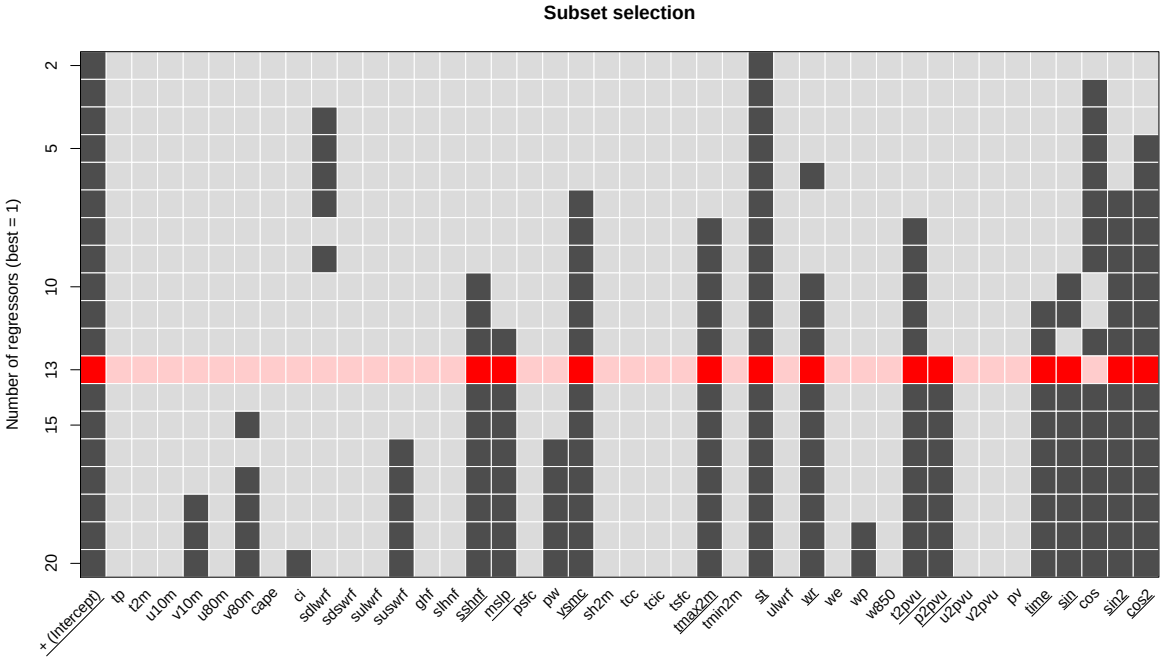

Figure 5: Subset selection for `ms3`.

it is striking that `t2m` is not considered for any of the subsets in `ms3` while we forced it into `ms2`. The decision to always include the deterministic season/trend regressors, however, is confirmed as some of these regressors are already selected for low subset sizes and all of them are selected from size 14 onwards.

Nevertheless, the differences between `m2` and `m3` in terms of model fit are fairly small compared to the reference models `m0` and `m1`. Comparing the BIC and the root mean squared error (RMSE) gives

```
R> BIC(m0, m1, m2, m3)

   df      BIC
m0  7 9877.073
m1  8 9537.650
m2 14 9031.992
m3 14 9025.267

R> sqrt(sapply(list(m0, m1, m2, m3), deviance)/nrow(IbkTemperature))

[1] 3.602371 3.274711 2.814703 2.809505
```

showing that `m2` and `m3` improve the model substantially over `m0` and `m1`. By construction the BIC of `m3` is lower than that of `m2` but not by much.

In summary, this shows that best-subset selection can easily identify relevant variables beyond the direct model output for MOS regressions, yielding substantial improvements in forecasting quality. In a full meteorological application this should, of course, be further tested using cross-validation or other out-of-sample assessments. But as this is beyond the scope of this paper we confine ourselves to the in-sample assessment presented here.

To conclude, it is also worth pointing out that recently there has been increasing interest in MOS models beyond least-squares linear regression. Incorporating heteroscedasticity is important ensemble MOS models as well as censoring or truncation for quantities like precipitation or wind (see the **crch** package of Messner *et al.* 2016a for some examples). All-subsets regression for those models would be much more burdensome and is not available in **lmSubsets**. An alternative solution in those situations is for example boosting as proposed by Messner *et al.* (2016b).

5. Case study

`TODO: by Ana`

6. Benchmark

`TODO: Achim: Below are some rough ideas...Before going into detail about this one should check what other authors in this field have used for setting up their regressions. Following their templates might make the design easier and more convincing for the reviewers.`

Data-generating process:

- Allow increasing M and/or N.
- A certain fixed fraction of variables could be relevant (say 20%).
- All regressors could have the same distribution, e.g., standard normal or uniform on $[-1, 1]$.

- The relevant variables could also all have the same coefficient or follow a linearly decreasing pattern for example. All irrelevant variables have coefficient zero.
- Then one could consider two values of M (say, 200 and 1000) and a sequence of Ns (say, 20, 40, ...).

Competitors:

- lmSubsets and lmSelect, either exactly or with certain relaxations.
- Other exact solutions (at least: leaps).
- Approximate solutions (at least: subselect and/or glmulti).
- Penalized solutions (at least: glmnet).

Outcome measurements:

- Computation time of the R functions (especially for exact solutions).
- Proportion of correctly selected regressors.
- RSS, AIC, BIC.

Acknowledgements

This work was in part supported by the *Förderverein des wirtschaftswissenschaftlichen Zentrums der Universität Basel*, through research project B-123.

The authors are grateful to Jakob Messner for sharing the GEFS forecast data in `IbkTemperature`.

References

Akaike H (1974). "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control*, **19**(6), 716–723. doi:10.1109/tac.1974.1100705.

Bauer P, Thorpe A, Brunet G (2015). "The Quiet Revolution of Numerical Weather Prediction." *Nature*, **525**(7567), 47–55. doi:10.1038/nature14956.

Calcagno V, de Mazancourt C (2010). "**glmulti**: An R Package for Easy Automated Model Selection with (Generalized) Linear Models." *Journal of Statistical Software*, **34**(12), 1–29. doi:10.18637/jss.v034.i12.

Chambers JM, Hastie TJ (eds.) (1992). *Statistical Models in S*. Chapman & Hall, London.

Duarte Silva AP (2001). "Efficient Variable Screening for Multivariate Analysis." *Journal of Multivariate Analysis*, **76**, 35–62. doi:10.1006/jmva.2000.19202.

Elff M (2016). ***memisc**: Tools for Management of Survey Data and the Presentation of Analysis Results*. R package version 0.99.7-1, URL https://CRAN.R-project.org/package=memisc.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1), 1–22. doi:10.18637/jss.v033.i01.

Gatu C, Kontoghiorghes EJ (2003). "Parallel Algorithms for Computing All Possible Subset Regression Models Using the QR Decomposition." *Parallel Computing*, **29**(4), 505–521.

Gatu C, Kontoghiorghes EJ (2006). "Branch-and-Bound Algorithms for Computing the Best Subset Regression Models." *Journal of Computational & Graphical Statistics*, **15**, 139–156. doi:10.1198/106186006x100290.

Glahn HR, Lowry DA (1972). "The Use of Model Output Statistics (MOS) in Objective Weather Forecasting." *Journal of Applied Meteorology*, **11**(8), 1203–1211. doi:10.1175/1520-0450(1972)011<1203:TUOMOS>2.0.CO;2.

Golub GH, Van Loan CF (1996). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, 3rd edition. Johns Hopkins University Press, Baltimore, Maryland.

Hamill TM, Bates GT, Whitaker JS, Murray DR, Fiorino M, Jr TJG, Zhu Y, Lapenta W (2013). "NOAA's Second-Generation Global Medium-Range Ensemble Reforecast Data Set." *Bulletin of the American Meteorological Society*, **94**(10), 1553–1565. doi:10.1175/BAMS-D-12-00014.1.

Hastie TJ, Tibshirani RJ, Friedman J (2001). *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag, New York.

Hofmann M, Gatu C, Kontoghiorghes EJ (2007). "Efficient Algorithms for Computing the Best Subset Regression Models for Large-Scale Problems." *Computational Statistics & Data Analysis*, **52**, 16–29. doi:10.1016/j.csda.2007.03.017.

Hofmann M, Gatu C, Kontoghiorghes EJ, Zeileis A (2016). ***lmSubsets**: Variable Subset Selection in Linear Regressions*. R package version 0.1-0, URL https://CRAN.R-project.org/package=lmSubsets.

Kontoghiorghes EJ (2000). *Parallel Algorithms for Linear Models: Numerical Methods and Estimation Problems*, volume 15 of *Advances in Computational Economics*. Kluwer Academic Publishers, Boston. doi:10.1007/978-1-4615-4571-2.

Lumley T, Miller A (2009). ***leaps**: Regression Subset Selection*. R package version 2.9, URL https://CRAN.R-project.org/package=leaps.

McLeod AI, Xu C (2014). ***bestglm**: Best Subset GLM*. R package version 0.34, URL https://CRAN.R-project.org/package=bestglm.

Messner JW, Mayr GJ, Zeileis A (2016a). "Heteroscedastic Censored and Truncated Regression with **crch**." *The R Journal*. Forthcoming, URL https://journal.R-project.org/archive/accepted/messner-mayr-zeileis.pdf.

Messner JW, Mayr GJ, Zeileis A (2016b). "Non-Homogeneous Boosting for Predictor Selection in Ensemble Post-Processing." *Working Paper 2016-04*, Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. URL http://EconPapers.RePEc.org/RePEc:inn:wpaper:2016-04.

Miller AJ (2002). *Subset Selection in Regression*, volume 95 of *Monographs on Statistics and Applied Probability*. 2nd edition. Chapman and Hall. doi:10.1201/9781420035933.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**(2), 461–464. doi:10.1214/aos/1176344136.

Smith DM, Bremner JM (1989). "All Possible Subset Regressions Using the QR Decomposition." *Computational Statistics & Data Analysis*, **7**(3), 217–235. doi:10.1016/0167-9473(89)90023-6.

Wolters MA (2015). "A Genetic Algorithm for Selection of Fixed-Size Subsets with Application to Design Problems." *Journal of Statistical Software, Code Snippets*, **68**(1), 1–18. doi:10.18637/jss.v068.c01.

Affiliation:

Marc Hofmann
TODO: address