A graph approach to generate all possible regression submodels

Cristian Gatu\textsuperscript{a,d,*}, Petko I. Yanev\textsuperscript{b,e}, Erricos J. Kontoghiorghes\textsuperscript{c,f}

\textsuperscript{a}Institut d’informatique, Université de Neuchâtel, Switzerland
\textsuperscript{b}INRIA/IRISA, Rennes, France
\textsuperscript{c}Department of Public and Business Administration, University of Cyprus, Cyprus
\textsuperscript{d}Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
\textsuperscript{e}Faculty of Mathematics and Informatics, University of Plovdiv “Paisii Hilendarski,” Bulgaria
\textsuperscript{f}School of Computer Science and Information Systems, Birkbeck College, UK

Available online 27 February 2007

Abstract

A regression graph to enumerate and evaluate all possible subset regression models is introduced. The graph is a generalization of a regression tree. All the spanning trees of the graph are minimum spanning trees and provide an optimal computational procedure for generating all possible submodels. Each minimum spanning tree has a different structure and characteristics. An adaptation of a branch-and-bound algorithm which computes the best-subset models using the regression graph framework is proposed. Experimental results and comparison with an existing method based on a regression tree are presented and discussed.

Keywords: Graphs; Regression trees; Model selection; Combinatorial algorithms

1. Introduction

An important topic in statistical modeling is that of subset-selection regression which is a matter of finding the best regression equation (Hastie et al., 2001; Hocking, 1976, 1983; Hocking and Leslie, 1967; LaMotte and Hocking, 1970). Given a list of possible variables to be included in the regression, the problem is to select a subset which optimizes some statistical criterion. The latter originates in the estimation of the corresponding submodel (Miller, 1984, 2002; Searle, 1971; Seber, 1977). In the case of the standard regression model with \( n \) parameters, there are \( 2^n - 1 \) possible submodels that have to be compared. Estimating the parameters of a Vector Autoregressive (VAR) process is another important case where subset models need to be specified. A \( G \)-multivariate VAR process of order \( p \) yields \( 2^{pG^2} - 1 \) submodels (Gatu and Kontoghiorghes, 2005). It is convenient to enumerate and evaluate all possible submodels when the number of parameters to choose from is not too large (Miller, 2002; Sen and Srivastava, 1990). The advantage of such an exhaustive search is that it is guaranteed to yield the optimum solution.

* Corresponding author. Institut d’informatique, Université de Neuchâtel, Switzerland.

\textit{E-mail addresses:} cristian.gatu@unine.ch (C. Gatu), petko.yanev@irisa.fr (P.I. Yanev), erricos@dcs.bbk.ac.uk (E.J. Kontoghiorghes).

0167-9473/S - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2007.02.018
Strategies for generating all possible subset regression models have been previously considered (Clarke, 1981; Gatu and Kontoghiorghes, 2003; Smith and Bremner, 1989). A dropping-columns algorithm that generates a regression tree together with a version of the algorithm that works in parallel has been proposed. These methods entail the QR factorization of the matrix of the regressors followed by its re-triangularization after the deletion of some of its columns.

Once a submodel comprising the variables \( \{v_1, v_2, \ldots, v_d\} \) is derived via a regression tree, the submodels corresponding to the subsets \( \{v_1\}, \{v_1, v_2\}, \ldots, \{v_1, v_2, \ldots, v_d\} \) become available without any extra computational cost. These subsets correspond to the already triangular \( 1 \times 1, 2 \times 2, \ldots, d \times d \) leading principal submatrices (Gatu and Kontoghiorghes, 2003, 2006a). Thus, the problem of generating all \( 2^n - 1 \) possible subset regression models becomes equivalent to that of generating the regression tree which has the fully specified model \( \{1, 2, \ldots, n\} \) as its root vertex.

Recently, an algorithm for computing the main matrix factorizations arising in the estimation of seemingly unrelated regression equation models with common variables has been introduced (Kontoghiorghes, 2000; Kontoghiorghes and Clarke, 1995; Yanev et al., 2004). This can be seen as equivalent to estimating a set of submodels with common variables.

The algorithm explores a weighted directed graph, which represents the possible sequences for deriving the submodels. The vertices represent the sets of variables defining the different models. An arc \( a \rightarrow b \) exists if and only if the set of variables representing vertex \( b \) is a subset of the set of variables representing vertex \( a \). The weight of the arc is defined by the computational cost for obtaining the submodel represented by vertex \( b \), given the already derived submodel in vertex \( a \). Thus, the shortest path from the root vertex to the remaining vertices defines the sequence which minimizes the computational cost of the successive estimation of the submodels.

A graph which represents all possible subset regression models is introduced. The graph is a generalization of the regression tree in Gatu and Kontoghiorghes (2003, 2005, 2006a). The regression tree describes one of the shortest paths for traversing the graph. Furthermore, the spanning trees of the graph provide all possible subset models with the same minimum computational complexity, which makes them equivalent.

The next section introduces a class of regression graphs which can be employed in statistical model selection. It describes how the combinatorial problem of enumerating all possible subset regression models can be formalized with directed graphs. Theoretical measures of the complexity of generating all models by traversing a regression graph are presented. The relationship between the regression graphs and the regression trees is displayed. It is revealed how the various minimum spanning (regression) trees (MST) can be drawn from the regression graph. The merits of the derived regression trees are discussed. Section 3 presents the generalization of a branch-and-bound algorithm (BBA) for computing the best-subset regression models using the graph structure. Finally, Section 4 provides the conclusions.

2. Subset model selection and regression graphs

Consider the standard regression model

\[
y = A\beta + \varepsilon, \quad \varepsilon \sim (0, \sigma^2 I_m),
\]

where \( A \in \mathbb{R}^{m \times n} \) comprises the variables (columns) \( v_1, \ldots, v_n \) and \( m > n \). The QR decomposition (QRD) of the full rank \( A \) is given by

\[
Q^TA = \begin{pmatrix} R & 0 \\ 0 & \ast \end{pmatrix}_{m-n} \quad \text{with} \quad Q^T y = \begin{pmatrix} \hat{y} \\ \tilde{y} \end{pmatrix}_{m-n}.
\]

Here, \( Q \in \mathbb{R}^{m \times m} \) is orthogonal and \( R \in \mathbb{R}^{n \times n} \) is upper triangular. The least-squares solution and the residual sum of squares (RSS) of (1) are given by \( \hat{\beta} = R^{-1}\hat{\gamma} \) and RSS = \( \hat{y}^T \tilde{y} \), respectively. A submodel of (1) can be expressed with a selection matrix \( S \) which comprises columns of the \( n \times n \) identity matrix \( I_n \). Consider the submodel

\[
y = A(S)\beta(S) + \varepsilon, \quad \varepsilon \sim (0, \sigma^2 I_m),
\]

where \( A(S) = AS \in \mathbb{R}^{m \times d}, \beta(S) = S^T\beta \in \mathbb{R}^d \) and \( 1 \leq d < n \). The QRD of \( A(S) \) is required for the estimation of submodel (3). This is equivalent to re-triangularizing \( R \) in (2) after deleting \( n - d \) columns (Kontoghiorghes, 2000;
The least-squares estimator of submodel (3) and its corresponding RSS are given by

\[
\hat{y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_d \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}
\]

where the set of vertices and the set of arcs are given, respectively, by

\[
V = \{v_1, v_2, \ldots, v_n\}
\]

and

\[
E = \{E_{i,j}: V_i, V_j \in V, V_j \subset V_i\}
\]

The structure of \(G_V\).

The directed graph—hereafter called regression graph—is denoted by

\[
G(V) = (\mathcal{V}, \mathcal{E}, \mathcal{U})
\]

where the set of vertices and the set of arcs are given, respectively, by \(\mathcal{V} = V\) and

\[
\mathcal{E} = \{E_{i,j}: V_i, V_j \in \mathcal{V}, V_j \subset V_i\}
\]

and \(\mathcal{U} = \{E_{i,j}: V_i, V_j \in V, V_j \subset V_i\}\).

**Fig. 1.** The graph \(G_V\) for \(V = \{1, 2, 3, 4\}\).
The weight of $E_{i,j}$ is denoted by $C_{i,j}$ and is proportional to the complexity of estimating the submodel corresponding to $V_j$ given that the one corresponding to $V_i$ has already been estimated. Recall that this is equivalent to computing (4), which re-triangularizes an upper-triangular matrix after deleting some columns (Gatu and Kontoghiorghes, 2003, 2006a; Golub and Van Loan, 1996; Kontoghiorghes and Clarke, 1993; Yanev et al., 2004). Let $k = |V_j|$, $V_j = \{v_{j1}', v_{j2}', \ldots, v_{jk}'\}$, the arc $E_{i,j}$ from some vertex $V_i$ towards $V_j$ exist and $p_{i,t}$ denote the position of the $v_{jt}'$ in $V_i$ ($t = 1, \ldots, k$). Note that, since $V_j \subset V_i$, $p_{i,t} \geq t$ for every $t$. Then, the weight of the arc $E_{i,j}$ is given by

$$C_{i,j} = \sum_{t=1}^{k} (p_{i,t} - t)(k - t + 1). \quad (6)$$

This corresponds to the number of elementary operations used in constructing and applying a sequence of Givens rotations on 2-row submatrices in order to annihilate the non-zero elements below the main diagonal of the submatrix corresponding to $V_j$ (Gatu and Kontoghiorghes, 2006a; Yanev et al., 2004). Fig. 2 shows this sequence of Givens rotations for $V_i = \{1, 2, 3, 4, 5, 6\}$ and $V_j = \{1, 2, 4, 5, 6\}$, that is, when the third variable has been deleted. Fig. 3 shows the weights $C_{i,j}$ corresponding to all the arcs $E_{i,j}$ of the vertex $V_i = \{1, 2, 3, 4\}$.

Now, using this graph representation, the problem of generating all $2^n - 1$ subsets of $\{1, 2, \ldots, n\}$ becomes equivalent to visiting the $2^{(n-1)}$ vertices of the graph $\mathcal{G}_{\{1,\ldots,n\}}$ (Fig. 1(a)). The arcs and weights of $\mathcal{G}_{\{1,\ldots,n\}}$ provide all the possibilities and the associated cost of moving from one vertex to another by deleting variables. Finally, in order to derive the triangular factors of all vertices with minimum cost, the optimal path for visiting all the vertices is required. This is equivalent to finding one or more MST of the proposed regression graph (Kruskal, 1956; Prim, 1957).

2.1. Derivation of the minimal regression graph

The vertices of $\mathcal{G}_V = (\mathcal{X}, \mathcal{Y})$ with $|V| = n$ can be partitioned into $n$ levels $L_1, \ldots, L_n$, such that $L_k$ contains all vertices having exactly $k$ variables, i.e.

$$L_k = \{W : W \in \mathcal{X} \text{ and } |W| = k\}, \quad k = 1, \ldots, n.$$ 

Note that $L_n = \{\{1, \ldots, n\}\}$ and $L_1 = \{\{n\}\}$. Now, let $n \geq p > q > r \geq 1$ and the vertices $V_x \in L_p$, $V_h \in L_q$ and $V_t \in L_r$, such that the arcs $E_{x,t}$ and $E_{h,t}$ exist. If there is a path from $V_x$ to $V_h$, then $C_{h,t} \leq C_{x,t}$ (Yanev et al., 2004). Therefore, the arc $E_{x,t}$ can be deleted from the graph. Note that a path from $V_x$ to $V_h$ exists if and only if $V_h$ is a subset of $V_x$.  

![Fig. 2. Re-triangularization of an $n \times n$ upper-triangular matrix after deleting the $i$th column using Givens rotations, where $n = 6$ and $i = 3$.](image)

![Fig. 3. The non-trivial vertices of $V_i$ and the corresponding weight arcs for $V_i = \{1, 2, 3, 4\}$.](image)
Fig. 4. The non-adjacent levels “arcs-deletion-rule” for \( \mathcal{V}_s = \{1, 2, 3, 4\} \), \( p = 4 \), \( \mathcal{V}_h = \{1, 4\} \), \( q = 2 \), \( \mathcal{V}_t = \{4\} \) and \( r = 1 \).

Following this rule, the graph \( \mathcal{G}_\mathcal{V} \) can be reduced by removing all the arcs between non-adjacent levels. The resulting graph is denoted by

\[
\mathcal{G}_\mathcal{V} = (\mathcal{X}, \mathcal{U}),
\]

where \( \mathcal{X} \) is the same set of vertices as of \( \mathcal{G}_\mathcal{V} \) and

\[
\mathcal{U} = \{E_{i,j} = (V_i, V_j) : E_{i,j} \in \mathcal{U} \text{ and } |V_i| - |V_j| = 1\}
\]

is the set of arcs. Fig. 5 illustrates the reduced graph \( \mathcal{G}_\mathcal{V} \) for \( \mathcal{V} = \{1, 2, 3, 4, 5\} \).
where \( \mathcal{X} \) is the same set of vertices as \( \mathcal{G}_V \) and \( \overline{\mathcal{G}}_V \), and

\[
\mathcal{E} = \{E_{i,j} : E_{i,j} \in \overline{\mathcal{G}}_V \text{ and } C_{i,j} = \min\{C_{k,j} : E_{k,j} \in \overline{\mathcal{G}}_V\}\}
\]

is the set of arcs.

Fig. 7 illustrates the graph \( \Gamma_V \) after deleting the unnecessary arcs, for \( V = \{1, \ldots, n\} \) and \( n = 5 \). The (minimal) weights of the arcs are also displayed. Each spanning tree of \( \Gamma_V \) has the same total cost of visiting all the vertices which is equal to the sum of the weights of all its arcs. This total sum is optimal (minimum). Thus, \( \Gamma_V \) is a minimal graph—hereafter, minimal regression graph, i.e. each spanning tree is a MST.

The “arcs-deletion-rules” reduce the graph \( \mathcal{G}_V \) in (5) to \( \Gamma_V \). The latter can be also constructed directly using a recursive definition. Let \( |V| = 1 \), i.e. \( V = \{v\} \). In this case, \( \Gamma_V \equiv \mathcal{G}_V \). This is the graph having only one vertex corresponding to the set \( \{v\} \), and no arcs. Assume, now, that \( \Gamma_V = (\mathcal{X}, \mathcal{E}) \) has been defined for some set \( V = \{v_1, \ldots, v_n\} \) and a new variable \( w \) is added to \( V \). The definition of \( \Gamma_{w,V} \) is required, where \( w \cdot V \equiv \{w, v_1, \ldots, v_n\} \). Let

\[
\hat{\Gamma}_{w,V} = (\hat{\mathcal{X}}, \hat{\mathcal{E}}) \quad \text{with} \quad \hat{\mathcal{X}} = \{w \cdot V_i : V_i \in \mathcal{X}\}, \quad \hat{\mathcal{E}} = \{\hat{E}_{i,j} = (w \cdot V_i, w \cdot V_j) : E_{i,j} = (V_i, V_j) \in \mathcal{E}\}.
\]

The weight of \( \hat{E}_{i,j} \) is the same as that of \( E_{i,j} \) for all \( E_{i,j} \in \mathcal{E} \). Furthermore, the graphs \( \Gamma_V \) and \( \hat{\Gamma}_{w,V} \) have the same size and structure. The only difference is that the vertices represent different subsets (models) since the vertices of \( \hat{\Gamma}_{w,V} \) are obtained by adding \( w \) to each vertex of \( \Gamma_V \). Both graphs \( \Gamma_V \) and \( \hat{\Gamma}_{w,V} \) are shown in Fig. 8(a) and (b), respectively.
for $V = \{2, 3, 4, 5\}$ and $w = 1$. Notice that the variable $w$ exists only in $\widehat{T}_{w,V}$ but not in $\Gamma_V$. In Fig. 8(a) the graph has been stretched by prolonging the arcs and the recursive subgraphs $\Gamma_{[3,4,5]}$ and $\widehat{T}_{2,[3,4,5]}$ have been framed in order to illustrate its recursive definition.

Now, given $\Gamma_V$ and $\widehat{T}_{w,V}$, the graph $\Gamma_{w,V}$ is defined as

$$
\Gamma_{w,V} = (\overline{X}, \overline{E}) \quad \text{with} \quad \overline{X} = \overline{X} \cup \widehat{X}, \quad \overline{E} = \overline{E} \cup \widehat{E} \cup \{E^*_{i,i} = (\{w, v_1, \ldots, v_n\}, \{v_1, \ldots, v_n\}); i = 1, \ldots, n\}. \quad (7)
$$

The weight $C^*_{i,i}$ of the added arcs of form $E^*_{i,i}$ that connect the two subgraphs $\Gamma_V$ and $\widehat{T}_{w,V}$ is given by $C^*_{i,i} = i(i + 1)/2$. This completes the recursive definition of $\Gamma_V$. From the recursive definition in (7) and the computation of the weight of the added arcs $E^*_{i,i}$ ($i = 1, \ldots, n$) in (6) it follows that each graph $\Gamma_{w,V}$ can be constructed once the smaller graph $\Gamma_V$ is derived. Fig. 9 shows an example of the graph $\Gamma_{w,V}$ for $w = 1$ and $V = \{2, 3, 4, 5, 6\}$. The two subgraphs $\Gamma_V$ and $\widehat{T}_{w,V}$ are well distinguished at the left and right of the illustration, respectively. The weights of the new arcs that
The graph $\Gamma_{\{1,2,3,4,5\}}$.

The regression tree $T_5$.

Fig. 10. The MST $T_n$ derived from $\Gamma_{V}$, for $V = \{1, 2, \ldots, n\}$ and $n = 5$.

connects them are shown. Hence, the recursive weighted directed graph $\Gamma_{V}$ is optimal in the sense that all its spanning trees are MST and provide an optimal computational procedure (i.e. minimum computational cost) for deriving all possible submodels.

2.2. Minimum spanning trees and regression trees

The MSTs derived from $\Gamma_{V}$ differ in their structure. Thus, some of them exhibit properties and characteristics that could be more suitable for specific problems such as parallel strategies for deriving all subset models, branch-and-bound selection and subrange model derivation (Gatu and Kontoghiorghes, 2003, 2006a; Hofmann et al., 2007; Kontoghiorghes, 1999, 2000, 2005). Consider the MST of $\Gamma_{\{1,2,3,4,5\}}$, denoted by $T_5$ of Fig. 10(b). This tree is of particular interest because it keeps the recursive structure of the graph. That is, it can be recursively constructed independently of $\Gamma_{\{1,2,3,4,5\}}$. This regression tree, $T_n$, has been investigated and its properties thoroughly exploited within the context of model selection in Gatu and Kontoghiorghes (2006a). A parallel algorithm for computing all possible subset regression models using this regression tree has also been proposed (Gatu and Kontoghiorghes, 2003).

Another MST of $\Gamma_{\{1,2,\ldots,n\}}$, denoted by $T_n^*$, which can be of particular interest is presented in Fig. 11(b), where $n = 5$. Note that the cost of generating $T_n^*$ is the same as for that of generating $T_n$. However, $T_n^*$ has more balanced structure in the sense that the cost of generating the left subtrees with root vertices $\{1,2,4,5\}$ and $\{1,3,4,5\}$ is the
same as the cost of generating the right subtrees with root vertices \{2, 3, 4, 5\} and \{1, 2, 3, 5\}. This property allows \(T_n^*\) to be computed by two processors using a complete load distribution (Kontoghiorghes, 2000), i.e. two processors can compute the \(T_n^*\) in half the time needed for computing the whole tree by a single processor.

3. Branch-and-bound strategy for model selection

A BBA for computing the best-subset regression models together with its heuristic counterpart has recently been developed (Gatu and Kontoghiorghes, 2006a; Hofmann et al., 2007). Here, a generalization of these strategies using the minimal regression graph \(\Gamma_{V}\) is proposed (GBBA) (Brusco and Stahl, 2005; Furnival and Wilson, 1974; Lawler and Wood, 1966; Ridout, 1988). The pruning procedure previously described in detail can be enhanced by using the arcs of the graph (Gatu and Kontoghiorghes, 2006a; Hofmann et al., 2007). That is, if the cutting test holds, then the right-hand side subtrees (subgraphs) are cut, but also some subgraphs from the left-hand side of the graph.

The pruning is illustrated in Fig. 12 for \(n=6\). Specifically, using the regression tree approach (BBA), the vertex \{1, 6\} is tested against the bound vertex \{1, 3, 4, 5, 6\}. If the test holds, then the whole subtree having as root \{1, 3, 4, 5, 6\} is cut. Now, using the graph structure (GBBA), in particular the arc ((\{1, 3, 4, 5, 6\}, \{3, 4, 5, 6\}), the subgraph having as root the vertex \{3, 4, 5, 6\} can be also pruned, except of the vertices on the leftmost path—\{3, 4, 5, 6\}, \{4, 5, 6\}, \{5, 6\}, \{6\}—that contain submodels of size one—\{3\}, \{4\}, \{5\}, \{6\}. The latter vertices need to be tested before pruning. The reduced
The regression graph for 6-variable initial list.

The reduced search.

The details of the GBBA procedure are shown in Algorithm 1. A set $V$ with $k$ passive variables which are kept in the subsequent generated subgraphs is denoted by the pair $\{V, k\}$. These pairs represent the vertices of the graph and they are stored in the list $L$ based on the “first in–first out” principle. Initially, the root vertex of the graph—which corresponds to the initial full set of variables—is stored in the list $L$. The number of passive variables is set to 0, i.e. all variables are candidate to be dropped. During the execution of the algorithm a pair $\{V, k\}$ is extracted from the list and the leading $(|V| - k)$ new submodels are obtained. The remaining $k$ leading submodels have been already generated at the earlier stages of the algorithm. In subsequent steps, the children of the vertex are generated one by one if the branch-and-bound cutting test fails. A variable from the first set in the list $L$ is flagged for the deletion when a subgraph is pruned. In the above example (Fig. 12), the variable 2 of the vertex $\{2, 3, 4, 5, 6\}$ is flagged. This
Table 1
Execution times in seconds for the BBA and GBBA without and with preordering

<table>
<thead>
<tr>
<th>n</th>
<th>BBA</th>
<th>GBBA</th>
<th>BBA–1</th>
<th>GBBA–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>8.13e+00</td>
<td>7.73e+00</td>
<td>1.60e−01</td>
<td>1.50e−01</td>
</tr>
<tr>
<td>35</td>
<td>6.29e+01</td>
<td>5.96e+01</td>
<td>2.20e−01</td>
<td>2.10e−01</td>
</tr>
<tr>
<td>40</td>
<td>3.46e+02</td>
<td>3.28e+02</td>
<td>5.09e+00</td>
<td>5.05e+00</td>
</tr>
<tr>
<td>45</td>
<td>8.23e+03</td>
<td>7.98e+03</td>
<td>7.49e+00</td>
<td>7.35e+00</td>
</tr>
<tr>
<td>50</td>
<td>1.90e+04</td>
<td>1.87e+04</td>
<td>1.20e+01</td>
<td>1.15e+01</td>
</tr>
<tr>
<td>55</td>
<td>a</td>
<td>a</td>
<td>4.80e+02</td>
<td>4.65e+02</td>
</tr>
<tr>
<td>60</td>
<td>a</td>
<td>a</td>
<td>9.67e+02</td>
<td>9.60e+02</td>
</tr>
<tr>
<td>65</td>
<td>a</td>
<td>a</td>
<td>9.21e+03</td>
<td>9.03e+03</td>
</tr>
</tbody>
</table>

*Canceled after 24 h.*

allows the algorithm to cut directly the subgraph with root \{3, 4, 5, 6\} when processing the vertex \{2, 3, 4, 5, 6\}. The core operation of the algorithm is \text{drop}(V, i), which generates a new child vertex by deleting the \(i\)th element of \(V\) and computing the QRD (4). The output of the Algorithm 1 is given by \(\rho_j (j = 1, \ldots, n)\), which are the residuals of the best submodels comprising \(j\) variables. That is,

\[
\rho_j = \min\{\text{RSS}(V) : V \subseteq \{v_1, \ldots, v_n\} \text{ and } |V| = j\}, \text{ for } j = 1, \ldots, n.
\]

The performance of BBA was significantly improved by sorting the variables in the initial list, prior the execution of the algorithm (Gatu and Kontoghiorghes, 2006a). The sorting criterion is given by the individual strength of each variable. The strongest variable is defined as the one that causes the biggest increase of the residuals when deleted from the model. The same strategy has been also applied on the GBBA. These versions are denoted by BBA–1 and GBBA–1, respectively. The algorithms have been implemented in C++ with BLAS and LAPACK using GNU compiler collection on Pentium class machines with 512 MB of RAM, running under Linux. The data sets employed in the experiments have been constructed using randomly generated data from a uniform distribution. Table 1 and Fig. 13 show the execution times in seconds and the log\(_2\) scaled number of generated vertices of the BBA, GBBA, BBA–1 and GBBA–1, respectively. All methods are exhaustive and generate the (same) best submodels.
Algorithm 1. The graph BBA for finding the best-subset models

```
procedure gbba(V_root)
  ρ_j ← ∞, for j = 1, ..., |V_root|
  insert {V_root, 0} in the list L  [List L uses the “first in–first out” principle]
  while L ≠ ∅ do
    extract {V, k} from the list L, where n = |V| and V = {v_1, v_2, ..., v_n}.
    ρ_j ← min(ρ_j, RSS({v_1, ..., v_j})), for j = k + 1, ..., n
    if n - k < 1 then
      if RSS(V) > ρ_{k+1} then
        if L ≠ ∅ then
          extract {W, μ} from the list L
          flag the variable w_{μ+1+|W|-n} for deletion
          insert {W, μ} in the list L
        end if
      else
        if variable v_{k+1} is flagged for deletion then
          W ← V
          repeat
            W ← drop(W, k + 1)
            ρ_j ← min(ρ_j, RSS({w_1, ..., w_j})), for j = k + 1, ..., |W|
            i ← |W| - k
            if RSS(W) > ρ_{k+1} then
              i ← 1  [Terminate the repeat-until iteration]
            end if
          until i = 1
        else insert {drop(V, k + 1), k} in the list L
      end if
    else insert {drop(V, j), j - 1} in the list L end if
  end while
end procedure
```

As in the case of the BBA, it can be observed that preordering the variables considerably improves the performance of GBBA. The graph approach performs slightly better when compared with the classical approach based on regression trees. In fact, it can be proven that the two algorithms are equivalent. That is, any left-hand side subgraph pruned by the GBBA is also cut by the BBA. The improvement observed by GBBA compared with the BBA is due to the fact that the former derives additional vertices in earlier stages (the shaded rectangular vertices in Fig. 12(b)) and, thus, provides better bounds when investigating the remaining subgraphs.

4. Conclusions

A directed-graph approach which can be employed in statistical model selection has been proposed. The combinatorial problem of generating all possible subset regression models has been formalized by means of the regression graph.
Thus, enumerating all subsets becomes equivalent to traversing all vertices of the graph. Theoretical measures of the complexity of generating all submodels by traversing the regression graph have been presented. The properties of the graph have been investigated and exploited in order to reduce significantly the number of arcs. The resulting graph has a recursive structure and all of its spanning trees are minimum spanning trees (MST). Thus, it provides an optimal computational procedure for generating all possible subset models. The MSTs yield different regression tree strategies for model selection. Each tree has different structure and characteristics.

A generalization of a BBA for computing the best-subset models using the regression graph structure (GBBA) has been proposed. The algorithm avoids the need to generate all vertices of the graph when searching for the best submodels by pruning non-optimal subgraphs. Experiments have shown a small improvement of the GBBA over the existing BBA based on a regression tree (Gatu and Kontoghiorghes, 2006a). Heuristic versions of the BBA that use a tolerance parameter when deciding to prune a subtree have been proposed previously. They relax the objective of finding the optimal solution in order to improve runtime performance. These strategies can be adapted to the regression graph structure (Gatu and Kontoghiorghes, 2006a; Hofmann et al., 2007).

The above model selection algorithms can be modified to deal with Vector Autoregressive subset model selection (Gatu and Kontoghiorghes, 2003, 2005, 2006b; Gatu et al., 2006). The regression graph provides a framework that can be extended to solve similar combinatorial problems such as $k$-fold cross validation and identification of influential data. In this case, the vertex indices represent observations, rather than variables (Belsley et al., 1980; Hofmann et al., 2006).

Acknowledgements

The authors are grateful to the guest-editor Moon Huh and two anonymous referees for their valuable comments and suggestions.

The work of the first two authors was carried out during the tenure of ERCIM fellowships. In addition, this work is in part supported by the Swiss National Science Foundation Grants 101412-105978, 200020-100116/1, PIOI1-110144 and PIOI1-115431/1, and the Cyprus Research Promotion Foundation Grant KY-IT/0906/09.

Appendix A. Case study on Hald data set

The graph approach to best-subset model selection is illustrated below on the Hald data set known also as cement data. The data set comprises four independent variables, one dependent variable and 13 observations. Details can be found in Draper and Smith (1998) and Woods et al. (1932).

Figs. A1–A5 show the main steps when generating the graph $G^{\{1,2,3,4\}}$ corresponding to the Hald data. Initially, the root vertex is computed and the four leading submodels are used to initialize the RSS look-up table that stores the current best solution. This is illustrated in Fig. A1. Specifically, Fig. A1(a) shows the root vertex. The leading submodels and their corresponding RSS are shown on the second and the third rows, respectively. The RSS look-up table is depicted in Fig. A1(b).

The derivation of the next three vertices is illustrated in Figure A2. Following the computation of the second vertex $\{2,3,4\}$ a better submodel of size 1 is found, i.e. the submodel $\{2\}$, and the current solution is updated with the new RSS value 906.34. Similarly, the submodel $\{1,2,4\}$ computed in the fourth step improves the current solution of size 3 with the new value 47.97. The better models found and their corresponding RSS values are shown in bold font.

![Fig. A1. First step in deriving the graph $G^{\{1,2,3,4\}}$ for the Hald data.](image)
Fig. A2. The graph $\Gamma_{\{1,2,3,4\}}$ for the Hald data after four steps.

Fig. A3. The graph $\Gamma_{\{1,2,3,4\}}$ for the Hald data after six steps.

Fig. A4. The graph $\Gamma_{\{1,2,3,4\}}$ for the Hald data after seven steps.

Fig. A3 corresponds to the computation of the vertices $\{3, 4\}$ and $\{1, 4\}$. The newly generated submodels do not improve the current best solution.
Table A1
Graph approach to best-subset model selection on the Hald data

<table>
<thead>
<tr>
<th>Step</th>
<th>Computed vertex</th>
<th>Model size 4</th>
<th>Model size 3</th>
<th>Model size 2</th>
<th>Model size 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1, 2, 3, 4}</td>
<td>47.86</td>
<td>48.11</td>
<td>57.90</td>
<td>1265.69</td>
</tr>
<tr>
<td></td>
<td>(1, 2, 3, 4)</td>
<td>(1, 2, 3)</td>
<td>(1, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>[2, 3, 4]</td>
<td></td>
<td></td>
<td></td>
<td>906.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[1, 3, 4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>[1, 2, 4]</td>
<td></td>
<td>47.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1, 2, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>[1, 4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>[3, 4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>[2, 4] (CUT)</td>
<td></td>
<td></td>
<td></td>
<td>883.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>[4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A2
Hald data: the best-subset models

<table>
<thead>
<tr>
<th>Best submodel</th>
<th>{1, 2, 3, 4}</th>
<th>{1, 2, 4}</th>
<th>{1, 2}</th>
<th>{4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSS</td>
<td>47.86</td>
<td>47.97</td>
<td>57.90</td>
<td>883.87</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.67</td>
</tr>
<tr>
<td>$R^2_{adj}$</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.64</td>
</tr>
<tr>
<td>$C_P$</td>
<td>5.00</td>
<td>3.02</td>
<td>2.68</td>
<td>138.73</td>
</tr>
<tr>
<td>$s^2$</td>
<td>5.98</td>
<td>5.33</td>
<td>5.79</td>
<td>80.35</td>
</tr>
</tbody>
</table>

Fig. A4 illustrates the case where the branch-and-bound pruning is effective. The cutting test $57.90 \equiv \rho_2 < \text{RSS}((2, 3, 4)) \equiv 73.81$, where $\rho_2$ is the current solution for model size 2 and the RSS((2, 3, 4)) is the RSS of submodel {2, 3, 4}. The pruned vertex has a square frame.

Fig. A5. The graph $\Gamma_{\{1,2,3,4\}}$ for the Hald data after eight steps.
The final graph $\Gamma^{(1,2,3,4)}$ and the best solution are depicted in Figs. A5(a) and (b), respectively. The last computed vertex provides submodel $\{4\}$, which improves the previous current solution of size 1 with the new optimal value 883.87.

Table A1 summarizes the step-by-step derivation of the graph $\Gamma^{(1,2,3,4)}$ and the evolution of the current solution. Only the improved solutions—the value of the RSS and the corresponding submodel—are shown. Table A2 shows the best submodel for each model size and the corresponding values of the RSS, $R^2$, adjusted $R^2$, Mallows $C_p$ and the residual mean squares, i.e. RSS, $R^2$, $R^2_{\text{adj}}$, $C_p$ and $s^2$, respectively.

References


