Econometric Model Selection With More Variables Than Observations

Jurgen A. Doornik
University of Oxford and
Oxford OX1 1NF, UK
March 16, 2009

Preliminary version

Abstract

Several algorithms for indicator saturation are compared and found to have low power when there are multiple breaks. A new algorithm is introduced, based on repeated application of an automatic model selection procedure (Autometrics, see Doornik, 2009) which is based on the general-to-specific approach. The new algorithm can also be applied in the general case of more variables than observations. The performance of this new algorithm is investigated through Monte Carlo analysis.

The relationship between indicator saturation and robust estimation is explored. Building on the results of Johansen and Nielsen (2009), the asymptotic distribution of multi-step indicator saturation is derived, as well as the efficiency of the two-step variance. Next, the asymptotic distribution of multi-step robust estimation using two different critical values (a low one at first) is derived. The asymptotic distribution of the fully iterated case is conjectured, as is the asymptotic distribution of reweighted least squares based on least trimmed squares (Rousseeuw, 1984), called RLTS here. This allows for a comparison of the efficiency of indicator saturation with RLTS.

Finally, the performance of several robust estimators and the new approach is studied in the presence of a structural break. When there are many irrelevant regressors in the model, the robust estimators break down while the new algorithm is largely unaffected.

1 Introduction

Automatic model selection methods based on general-to-specific modelling (‘automated Gets’) have shown their value in comparison to other approaches, see Hoover and Perez (1999), Hendry and Krolzig (1999), Doornik (2009), Doornik (2008). The starting point for such methods is the full model, using the entire initial information set. This initial model must be estimable, which seems to confine such methods to situations where there are fewer variables $k$ than observations $T$. Otherwise,
when \( T \geq k \), the textbook answer is that the model cannot be estimated. Contrary to this Hendry and Krolzig (2005) propose an approach that works as follows: partition the initial model in \( K \) randomly created sets of variables, do model selection on each set, select surviving variables and repeat until \( k \) is sufficiently small. Hendry and Krolzig (2004) propose to make the blocks somewhat smaller, and run model selection on all cross blocks \((K(K-1)/2 \text{ model selections})\). Neither paper provides evidence on the practical performance of these algorithms.

Indicator saturation in a location scale model could be considered the canonical example of more variables than observations. With an intercept and impulse dummy for every observation, this model has \( T + 1 \) regressors for a sample size \( T \). Hendry, Johansen and Santos (2008) analyze this case analytically, starting from a two-way split. The procedure works by applying model selection twice, once with the first half of the dummy variables in the model, and once with the second half. Finally the selected dummies from both cases are added, and model selection is applied again to obtain the final model. They also provide Monte Carlo results under the null hypothesis that no dummy matters. Johansen and Nielsen (2009) formalize the split-sample indicator saturation procedure and extend the analytical results to more general models, allowing for stochastic regressors.

Santos (2008) considers the power of indicator saturation when there is a break in mean or variance in the location scale model. He uses a split half-way in the sample, at \( T/2 \). Monte Carlo evidence for the split-sample algorithm using a range of magnitudes for a break at \( 0.8T \) shows good power. However, this test is tailored towards a single break, and it is not clear if it would work well in the presence of multiple breaks (as required in a test for super exogeneity, Hendry and Santos, 2007). Indeed, we shall show severe loss of power for certain cases.

The objectives of this paper are

1. to make automatic general-to-specific model selection operational for indicator saturation, as well as the general case of more variables than observations,
2. to assess the performance of the new and previously proposed procedures using Monte Carlo methods,
3. to provide a comparison with some forward selection methods,
4. to provide a comparison with some robust estimation methods, and
5. to consider some new applications of the procedure.

Castle, Doornik and Hendry (2008) presents results of the performance of the proposed procedure in the presence of multiple breaks in a range of settings, and is therefore complimentary to the current paper.

2 Automatic model selection

In what follows, the automated Gets procedure can almost be considered as a ‘Black-box’: a final model is selected from the model that is formed from the initial set of candidate variables. The initial
model is called the *general unrestricted model* or GUM. Usually a set of terminal candidate models is found. In that case, an information criteria is used as the *tie-breaker*. In the block-search procedure we may instead select the final GUM, which is the union of the terminal candidate models.

The objective of automated Gets procedures is that the GUM is statistically well specified, which is verified through *mis-specification testing*. Then *diagnostic tracking* ensures that all terminal candidate models also satisfy these tests. A path *search* is used to simplify the GUM. Such a search is required to handle the complex intercorrelations that are commonly found in economic data. In *Autometrics*, which is used for all model selection results in this paper, this is a tree search (see Doornik, 2009). A simplification is excepted when the variabale that is removed is insignificant and when the new model is a valid reduction of the GUM. The latter requirement is called *encompassing the GUM or backtesting* and, in linear regression models, is based on the $F$-test of the removed variables.

The main choice in the application of *Autometrics* is the reduction $p$-value $p_a$, which is used for both backtesting and significance of individual regressors.

There are several mechanisms to avoid estimating $2^k$ models, again see Doornik (2009). This procedure has been found to be very effective: although the costs of statistical inference cannot be avoided (e.g. when selecting at 5\% for a variable that has a population $t$-value of two, we can only expect to detect it half the time), the costs of search are low (see Hendry and Krolzig, 2005).

Two automatic model selection methods that do not fit within the general-to-specific methodology are:

1. **Stepwise regression**: starting from the empty model, the omitted variable that would be most significant is added to the model. If at any stage variables in the model become insignificant, the most insignificant is deleted (so in each iteration up to one regressor can enter, and one can leave).\(^1\) This is repeated until all variables in the model are significant at $p_a$, and any omitted variable would be insignificant.

2. **Backward elimination**: all regressors are added to the initial model, then regressors are deleted one at a time starting from the least significant. This is continued until all regressors have a $p$-value of $p_a$ or less.

There are three important differences with automated Gets: (1) lack of search, (2) no backtesting, (3) no miss-specification testing/diagnostic tracking.

### 3 Indicator saturation and structural breaks

Consider the location-scale model:

\[ y_t = \mu + \epsilon_t, \quad \epsilon_t \sim \text{IN}[0, \sigma_\epsilon^2], \quad t = 1, \ldots, T. \]

\(^1\)This aspect differentiates step-wise regression from forward selection.
Writing $I_t$ for the impulse dummy that is 1 at period $t$, and 0 otherwise. The model that is saturated with impulse dummies:

$$y_t = \mu + \sum_{s=1}^{T} \delta_s I_s + \epsilon_t, \quad \epsilon_t \sim \text{IN}[0, \sigma^2_\epsilon], \quad t = 1, ..., T. \quad (1)$$

cannot be directly estimated by least squares. The split-sample algorithm works as follows.

1. Estimate using the first half of dummies (which is the same as estimating the mean and variance from the second half of the sample):

$$y_t = \mu + \sum_{s=1}^{\lfloor T/2 \rfloor} \delta_s I_s + \epsilon_t, \quad t = 1, ..., T.$$  

and apply model selection to give a set of dummies $I_1$.

2. Repeat for the remainder of the dummies:

$$y_t = \mu + \sum_{s=\lfloor T/2 \rfloor + 1}^{T} \delta_s I_s + \epsilon_t, \quad t = 1, ..., T.$$  

and apply model selection to give the second set of dummies $I_2$.

3. Regress $y_t$ on a mean, $I_1$, and $I_2$. Apply model selection to obtain the final model.

Figure 1 shows one sample of 100 observations from the following process:

$$y_t \sim \begin{cases} \text{IN}[\gamma, 1] & \text{for } t \leq \tau T, \\ \text{IN}[0, 1] & \text{for } t > \tau T. \end{cases} \quad (2)$$

With $\gamma = 10$ and the break set to $\tau = 0.75$. The second plot in Figure 1 shows the standardized residuals from the regression on an intercept only. They are all less than two (in absolute value): there are no outliers, despite the visually very clear pattern.

Figure 2 shows how the split-sample algorithm works when using Autometrics for the model selection. The first row of graphs shows the dummies included, then actual and fitted after model selection, and finally the dummies that were selected. The reason that so many dummies are selected here is not that they are significant, but caused by the diagnostic tracking that is part of any automated general-to-specific method. The second row shows the same for the second set of dummies. The bottom left graph shows the dummy that enter the dummies entering the final model prior to selection. This is now an estimable starting point, and only the relevant dummies survive selection.

In this example, using backward elimination in each sample split would give the same final result as Autometrics. The only difference is that in the first sample (block 1) no dummies would survive. Step-wise regression on the full sample would be very different though: as Figure 1n shows, no
Consider the following Monte Carlo experiment to compare the power of the three methods so far. The DGP is similar to (2), but now with mean zero at the start, and mean $\gamma$ after $\tau T$

$$\text{DGP:B } y_{1,t} = \gamma (I_{\tau T+1} + \ldots + I_T) + u_t, \quad u_t \sim N(0, 1).$$ (3)

Footnote: Here it is caused by the skewness component of the normality test, see Doornik and Hansen (2008).
Setting $T = 100$ and $\tau = 0.8$ means that twenty percent of the sample is in the break period. When $\gamma = 0$, the experiments are under the null of no break. The initial model consists of the $T$ dummies and an intercept, as in (1); note that the intercept is always included in all regressions:

$$
\text{MOD:B } y_{1,t} = \alpha + \sum_{i=1}^{T} \beta_i I_i + \varepsilon_t. \quad (4)
$$

Four estimation methods are compared

1. Step-wise regression on the full sample using selection $p$-value $p_a$, see §2. The intercept is included in all models, so the initial model is just a constant term.

2. The indicator saturation procedure introduced by Johansen and Nielsen (2009, §1.2.2.2). The precise implementation is described in §8.

3. Split-sample algorithm using a half-way split. Backward elimination is used in each of the two blocks. Next, an initial model is formed from the union of the dummies selected from each block, which forms the GUM for a normal Autometrics run. All models have an intercept, and the same $p$-value $p_a$ is used for the backward elimination and Autometrics.

4. Split-sample algorithm using a half-way split. Autometrics without diagnostic tracking is used in each of the two block. Next, an initial model is formed from the union of the dummies selected from each blocks, which forms the GUM for a normal Autometrics run. All models have an intercept, and the same reduction $p$-value $p_a$ is used throughout.

Switching diagnostic tracking off makes no difference to the selection performance here, but makes the procedure much faster. The reason is that the normality test can cause high retention of dummies in the first block, as shown in the example of Figure 2.

Reported are the gauge, which is the fraction of irrelevant variables in the final model, regardless of significance. In this case there are 80 irrelevant variables, so a gauge of 0.05 means that, on average, the final model contains 4 dummies that should not be there. The potency$^3$ records the fraction of relevant variables that are in the final model. A potency of 0.9 means that 18 correct dummies are in the final model (on average).

Step-wise regression at 5% is able to detect the correct dummies very well, but also includes much too many wrong dummies. At 1% it loses most power to detect the break, corresponding to the situation of Figure 1, where there are no outliers at the selected reduction $p$-value.

The two versions of the split-sample algorithm only differ in the model selection approach within the two blocks. Surprisingly, because the dummy variables are orthogonal, this does make quite a big difference when the information is relatively weak at $\gamma = 3$ with $p_a = 0.01$. Forcing an intercept in the model in that case also reduces the potency: if the constant is always omitted, we find 41% with method 3 (instead of 25.5) and 56% with method 4 (instead of 48.3).

$^3$Previously we called these size and power respectively.
Table 1: Comparison of step-wise regression, indicator saturation and two versions of the split-sample algorithm. $T = 100$ observations, $M = 1000$ replications, $p_a = 0.05, 0.01$. DGP:B has break in mean at 0.8$T$ of size $\gamma$. The initial model is the constant and $T$ dummies. Autometrics uses default settings, except when used in the split-sample blocks when diagnostic tracking is switched off.

These simulations suggest that the split-sample algorithm has good power to detect a single break in mean. The reason is that it is tailored for the situation of Table 1: the break is always in one half of the sample. So the dummies are essentially irrelevant in one block, while in the other there is a good match. There is not much sensitivity to the location of the break, provided that Autometrics is used throughout. This is illustrated in Table 2.4

When we allow for multiple breaks, it is less clear that the split-sample algorithm would work. Consider, for example, a break that comes and goes, symmetrically located around $T/2$. The potency can then be expected to be considerably lower. A three-way split would solve the problem, but, in general, we do not know the number of breaks nor their location.

The following experiment is designed to show that the split-sample algorithm cannot be seen as a general ‘pattern recognition’ algorithm. In DGP:S the section with the break is evenly spread across the data (as in seasonal data with one significant season). Then each half would effectively look the same, except for the smaller sample size. Defining $S = \lfloor (1 - \tau)^{-1} \rfloor$, the DGP is:

$$\text{DGP:S } y_{t_1}^{S_t} = \gamma (I_1 + I_{1+S} + I_{1+2S} + ...) + u_t, \quad u_t \sim N(0, 1).$$

---

4Forcing an intercept in all models does affect the potency: it is lower than when the constant is a free variable, sometimes quite substantially. However, the qualitative comparison between methods and for different choices of $\gamma$ or $\tau$ is unaffected. The benefit of forcing the constant term is that it does not feature in the gauge or potency.
Table 2: Experiment DGP:B with breaks of size $\gamma$ after $0.6T, 0.7T, 0.8T, 0.9T$. $T = 100$ observations, $M = 1000$ replications, $p_a = 0.01$.

So for $T = 100$ and $\tau = 0.8$, the mean is shifted by $\gamma$ at observations $t = 1, 6, 11, \ldots, 96$. This is 20% of the observations, just as for DGP:B with $\tau = 0.8$.

Table 3 shows that no method has any ability to detect the ‘seasonal’ pattern. The contrast with the single break experiment is quite striking.

In general, step-wise regression is very poor in almost all experiments, and backward elimination is no match for Autometrics (confirming the results in Doornik, 2008). In addition, the results show that the split-sample and indicator saturation algorithms are of quite limited use: they are optimized for a single break; also, it is unclear how to extend them when there are additional correlated variables.
Table 3: Comparison of step-wise regression, indicator saturation and three versions of the split-sample algorithm. \( T = 100 \) observations, \( M = 1000 \) replications, \( p_a = 0.01 \). DGP:B has break in mean at \( 0.8T \) of size \( \gamma \), DGP:S has the change in mean evenly spread at \( t = 1, 6, 11, \ldots \), also 20% of the sample.

4 More variables than observations

Even without indicator saturation, a researcher can be faced with a situation of more variables than observations. Allowing for a reasonable lag-length in a dynamic model can contribute to this.

The candidate approach in this case would be the cross-block algorithm proposed in Hendry and Krolzig (2004):

1. partition the set of \( k \) variables in \( K \) blocks,
2. apply automated Gets to each combination of blocks: this leads to \( K[K - 1]/2 \) runs of the model selection algorithm,
3. take the union of surviving variables as the new initial model,
4. if the dimension of this union is small enough, then stop, otherwise restart with the new set.

With three blocks, \( A, B \) and \( C \) this can be visualized as follows:

<table>
<thead>
<tr>
<th>DGP:B at 1% nominal size</th>
<th>DGP:S at 1% nominal size</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gamma = 3 )</td>
<td>( \gamma = 3 )</td>
</tr>
<tr>
<td>( \gamma = 4 )</td>
<td>( \gamma = 4 )</td>
</tr>
<tr>
<td>1. <strong>Step-wise regression</strong></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.1 0.0</td>
</tr>
<tr>
<td>Potency %</td>
<td>9.5 12.0</td>
</tr>
<tr>
<td>2. <strong>Indicator saturation (Johansen and Nielsen, 2009)</strong></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.5 0.7</td>
</tr>
<tr>
<td>Potency %</td>
<td>49.4 88.1</td>
</tr>
<tr>
<td>3. <strong>Backward elimination in blocks, then Autometrics</strong></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.3 0.6</td>
</tr>
<tr>
<td>Potency %</td>
<td>25.5 82.5</td>
</tr>
<tr>
<td>4. <strong>Autometrics(^*) in blocks followed by Autometrics</strong></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.3 0.4</td>
</tr>
<tr>
<td>Potency %</td>
<td>48.3 86.6</td>
</tr>
<tr>
<td>5. <strong>Autometrics in blocks followed by Autometrics</strong></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.3 0.4</td>
</tr>
<tr>
<td>Potency %</td>
<td>49.6 86.0</td>
</tr>
</tbody>
</table>

\(^*\) diagnostic tracking switched off
The main practical consideration would be how to choose the size of the blocks. In addition, automated Gets may find multiple terminal models. In that case, we could select just one (using the Schwarz Criterion, e.g.) or the union of the final models.

The experiments in this section are based on models 7 and 8 from Hoover and Perez (1999), denoted HP7 and HP8 respectively. Hoover and Perez construct a database consisting of quarterly observations on macro-economic variables, where unit roots are removed by differencing. The DGP for Hoover–Perez experiments 7 and 8 is:

**HP7:**
\[
y_{7,t} = 0.75y_{7,t-1} + 1.33x_{11,t} - 0.9975x_{11,t-1} + 6.44u_t, u_t \sim N[0,1], \quad R^2 = 0.58
\]

**HP8:**
\[
y_{8,t} = 0.75y_{8,t-1} - 0.046x_{3,t} + 0.0345x_{3,t-1} + 0.073u_t, u_t \sim N[0,1], \quad R^2 = 0.93
\]

HP7 has \(R^2 = 0.58\), and HP8 has \(R^2 = 0.93\); all coefficients have very high t-values (in excess of 8).

The first set of experiments start model selection from an initial model with the 3 variables that matter (the DGP) and an additional 37 that are irrelevant. The irrelevant variables consist of \(y_{t-2}, \ldots, y_{t-4}\) and another 17 regressors from the macro-economic database up to the first lag. A constant is always forced into the model. The sample size is \(T = 139\), and \(M = 1000\) Monte Carlo replications are used throughout.

Next, we create versions that have more variables than observations by adding 10 \text{IN}(0,1)\ regressors \(z_1, \ldots, z_{10}\) are to the initial model. Moreover, all regressors are added up to lag four, making 145 regressors in total. As before, only 3 matter, and the constant is always included. These experiments are labelled HP7big and HP8big in Table 4.

<table>
<thead>
<tr>
<th>HP7</th>
<th>HP8</th>
<th>HP7big</th>
<th>HP8big</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1. Stepwise</td>
<td>2. Stepwise</td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.9</td>
<td>0.9</td>
<td>1.7</td>
</tr>
<tr>
<td>Potency %</td>
<td>100.0</td>
<td>99.9</td>
<td>50.9</td>
</tr>
<tr>
<td>Time (minutes)</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>&lt;1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Autometrics</td>
<td>4. Cross-blocks</td>
</tr>
<tr>
<td>Gauge %</td>
<td>2.2</td>
<td>4.2</td>
<td>0.8</td>
</tr>
<tr>
<td>Potency %</td>
<td>99.1</td>
<td>59.1</td>
<td>99.9</td>
</tr>
<tr>
<td>Time (minutes)</td>
<td>5</td>
<td>206</td>
<td>54.1</td>
</tr>
</tbody>
</table>

Table 4: \(T = 139\), \(M = 1000\), \(p_a = 0.01\). HP7 and HP8 have 3 relevant and 37 irrelevant variables. HP7big and HP8big have 3 relevant and 141 irrelevant variables. Time is in minutes for all \(M = 1000\) runs.

First consider the columns labelled HP7 and HP8 in Table 4, reporting the results for the small experiments for step-wise regression and Autometrics. It has already been established beyond any
doubt that step-wise regression can be an inferior method for model selection. Despite this, there are some occasions where it works. HP7 is such a case: step-wise regression outperforms Autometrics in about 5% of the time. But HP8 shows that the extra time that Autometrics takes is a price well worth paying: the relevant variables are always found.

For the experiments with $k > T$ we compare step-wise regression with a cross-block algorithm using Autometrics (with diagnostics testing) for the blocks. The data set is divided in 6 blocks and we take the union of final models from each block into the next round. The results for the cross-block algorithm are ambivalent at best. For HP8big, on the one hand, the 100% success rate is maintained, at the cost of computational time. But this is not the case for HP7big: there is quite a deterioration relative to step-wise regression. At the same time, it is 500 times slower.\(^5\)

There are two drawbacks to the cross-blocks algorithm. The first is that there is no sense of learning: the algorithm restarts from scratch every time with a smaller set of candidate variables. Sometimes the algorithm struggles and progress is slow, because the set is only reduced with a few variables at a time.

Sofar we have established that neither the split-sample nor the cross-block algorithm is satisfactory. The next section introduces a new algorithm that performs better than either.

## 5 Autometrics block-search algorithm

To improve on the split-sample and cross-blocks algorithms, it seems useful to introduce a more progressive search: as the algorithm proceeds it learns about the variables that (may) matter. Because there are too many variables, there will be an alternation between specific-to-general and general-to-specific. This cannot be avoided, and will introduce a certain degree of arbitrariness in the procedure.

Inside the Autometrics block-search algorithm the set of all variables is partitioned in two: those that are currently selected (the candidate set) and the rest (the excluded set). The set of variables that is not currently selected is partitioned in blocks. Then two steps are alternated in succession:

1. **Expansion step**: partition the excluded set in blocks, then run over all blocks, searching for omitted variables. Make sure that the total number of omitted variables does not overwhelm the set of selected variables: together there should be less than $\alpha T$ variables.

2. **Reduction step**: find a new candidate set by running automated Gets on the current selection augmented by the omitted variables.

Figure 3 formalizes the expansion-reduction aspect of the algorithm, showing that, while this is iterated, there is also an iteration inside the expansion step. This is to ensure that the set of omitted variables is not too big.

There is one reduction $p$-value $p_a$ which governs all model selections, both inside the expansion step and for the reduction step. The reduction is always run at $p_a$, but the value may be varied if the

\(^5\)Timings are using Ox 5.1 Professional, Doornik (2007), on a quad core Intel Q6600, restricting Ox to one core only.
Definitions:
- \( \mathcal{B} \): set of all \( k \) regressors in initial unidentified GUM,
- \( S_i \): candidate set: regressors selected at end of iteration \( i \),
- \( \mathcal{B}\setminus S_i \): excluded set: regressors not selected at start of iteration \( i + 1 \),
- \( B_0^0, B_1^0, \ldots, B_B^0 \): block partitioning of \( B^0 = \mathcal{B}\setminus S_i \),
- \( O_i \): candidate omitted regressors in iteration \( i \).

Expansion step searches the excluded set for omitted variables:
1. Run \( B^0 \) reductions \( B_k^0 \cup S_i, \ldots, B_B^0 \cup S_i \) keeping \( S_i \) fixed;
2. Let \( B^1 \) be the selected regressors from all \( B_k^0, k = 1, \ldots, B^0 \);
3. Stop if \( \dim B^1 \) small enough, else restart using \( B^1 \) blocks for \( B^1 \) (if necessary shrinking \( p \)-value).
4. Upon convergence after \( J \) steps: \( O_i = B^J \)

Reduction step finds a new candidate set:
1. Find \( S_{i+1} \) from model selection on \( S_i \cup O_i \).

Figure 3: Expansion-reduction stages

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>starts from an empty model, and is run only once.</td>
</tr>
<tr>
<td>( B )</td>
<td>is run until convergence, using ( p_a ) for the expansion step.</td>
</tr>
<tr>
<td>( C )</td>
<td>is run only once, using ( 2p_a ) for the expansion step.</td>
</tr>
<tr>
<td>( D )</td>
<td>starts with ( 4p_a ) for the expansion step, using ( p_a ) thereafter. When stage ( D ) convergences, the block search terminates.</td>
</tr>
</tbody>
</table>

Table 5: Stages of the block-search algorithm.

The reason is that the reduction step must be identified. In addition, a temporary increase in the reduction \( p \)-value is used to increase the sensitivity of the algorithm: we may be searching for the proverbial needle in a haystack. The cost is that it may slightly increase the danger of overfitting. For this purposes four different stages in the overall procedure are identified, see Table 5.

Figure 4 shows how the expansion-reduction steps are embedded in the overall algorithm. The outside iteration counter is called \( i \), and the algorithm starts with the empty set as the initial candidate set. Stages are labelled \( A, B, C, D \) as noted above. Convergence occurs when the union of all variables that are, or have been, in the candidate set is unchanged from the previous iteration. So convergence is not based on fit: the likelihood will in general not be monotonically increasing. But
Set $i = 0$, $S_{-1} = \emptyset$, stage = $A$.

1. **Expansion step** to find $O_i$;
2. **Reduction step** to find $S_i$;
3. **Change the stage**:
   - 3a. if stage is $A$ set stage = $B$ and go to **Continuation**;
   - 3b. if stage is $C$ set stage = $D$ and go to **Continuation**;
   - 3c. else go to **Convergence** check.
4. **Convergence** if $S_0 \cup \cdots \cup S_i = S_0 \cup \cdots \cup S_{i-1}$ then
   - 4a. if stage is $B$ increment it to $C$,
   - 4b. else terminate block search.
5. **Continuation** increment $i$ and return to Step 1.

Figure 4: Outline of Autometrics block-search algorithm

then we do need to guard against cycling (i.e. a variable enters, then leaves, then enters again).

Some aspects of the algorithm have not been specified yet, in particular the block size and how to shrink the omitted variable set:

- **Block partitioning in expansion step**
  Write $k_i = \dim S_{i-1}$, the number of regressors selected when iteration $i$ starts. Let $k_f$ denote the regressors that are always forced to enter (e.g. the modeller could force an intercept in all models, in which case $k_f = 1$). The block size $k^b_i$ for the reduction step of iteration $i$ is then:
  $$k^b_i = \max(0, k_i - c_0/4, 2),$$
  where $c_0 = \min(128, 0.4[T - k_f])$.

Blocks are constructed in such a way that lags of the same variable are kept together (as far as possible, the actual block size to achieve this is allowed to go up to $1.5k^b_i$).

Note that, in general, the overall algorithm will be sensitive to the ordering of the variables.

- **Reduction $p$-value in expansion step**
  Table 5 shows that the expansion step is run at $s.p_a$, where $s = 1$ except in stage $C$ and at the start of stage $D$. In addition $s$ is varied inside the expansion step if necessary.

When very few variables are found in the first search for omitted variables in stage $A$ and the first three iterations of $B$, then $s$ is multiplied by 8. This is only allowed to happen once, and
introduces some bias against an empty model. The benefit is that it prevents the algorithm from aborting too early with very few variables found.

The number of omitted variables detected, \( k^o_j \), in the expansion step is forced to be no more than \( c_i \) and such that there at least 10 degrees of freedom in the initial model of the subsequent reduction step. When too many variables are found \( s \) is reduced as follows (using \( j \) for the iteration counter inside the expansion step):

\[
\begin{align*}
\text{if } k^o_j & \geq 4c_0 : \quad s_{j+1} = \min(s_j/16, 1); \\
\text{if } k^o_j & \geq 3c_0 : \quad s_{j+1} = \min(s_j/8, 1); \\
\text{if } k^o_j & \geq 2c_0 : \quad s_{j+1} = \min(s_j/4, 1); \\
\text{else} & \quad s_{j+1} = \min(s_j/2, 1).
\end{align*}
\]

- Backtesting with respect to the GUM is always used in expansion steps, but only for reduction steps in stages \( A \) and \( B \). The reason for the latter is that stages \( C \) and \( D \) start with increased \( p \)-values.

- Diagnostic tracking can have quite an impact on the speed of the expansion step. Moreover, when we start from the empty model, not much has been learned yet. Therefore diagnostic tracking is switched off in stage \( A \) and delayed if the current selection passes the test. It is always switched on in the reduction step, though.

- Tie-breaker When there are multiple terminal models, we always use the union of these (i.e. the final GUM) as the outcome of the model selection. There is a normal model selection run after the block algorithm has finished from the now identified ‘GUM’. This uses the Schwarz Criterion by default to choose the overall final model.

- Pre-search This is never used.

All these aspects could be changed, leading to slightly different behaviour of the algorithm. We found the current choices to work quite well, but it is likely that improvements (or perhaps simplifications) can be made.

6 Performance of the Autometrics block-search algorithm

Tables 6 and 7 show how the new Autometrics block-search algorithm, using expanding and contracting searches, improves on the others considered before. For experiment DGP:B, with the break for the last 20% of the observations, the block search behaves as well as the split-sample algorithm, but is very much faster. When the seasonal pattern is used, as in DGP:S, we had already seen that the split-sample algorithm has no power at all. The new algorithm performs much better — it is harder to detect these than a single break, but at \( \gamma = 4 \) the potency starts shooting up. Similarly, for the Hoover–Perez experiment HP7big, we see that the new algorithm has recovered the potency that automated Gets has in the small versions of the experiment.
Table 6: Comparison of the split-sample and the Autometrics block search algorithm. $T = 100, M = 1000, p_a = 0.01$. DGP:B has break in mean at $0.8T$ of size $\gamma$, DGP:S has the change in mean evenly spread at $t = 1, 6, 11, ...$.

<table>
<thead>
<tr>
<th></th>
<th>DGP:B at 1% nominal size</th>
<th>DGP:S at 1% nominal size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>$\gamma$</td>
<td>$\gamma$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5. <strong>Split-sample algorithm using Autometrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gauge %</td>
<td>0.3 0.4</td>
<td>0.5 2.8</td>
</tr>
<tr>
<td>Potency %</td>
<td>49.6 86.0</td>
<td>5.4 10.1</td>
</tr>
<tr>
<td>Time (minutes)</td>
<td>144 174</td>
<td>7 22</td>
</tr>
</tbody>
</table>

Table 7: Comparison of the cross-block and the Autometrics block search algorithm. $T = 139, M = 1000, p_a = 0.01$. HP7big and HP8big have 3 relevant and 141 irrelevant variables. Time is in minutes for all $M = 1000$ runs.

<table>
<thead>
<tr>
<th></th>
<th>HP7big</th>
<th>HP8big</th>
<th>HP7big</th>
<th>HP8big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-blocks</td>
<td>Gauge %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>0.8</td>
<td>1.4</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>Autometrics block search</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Potency %</td>
<td>59.1</td>
<td>96.9</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>Time (minutes)</td>
<td>206</td>
<td>99.9</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>

Finally, Table 8 shows that for lower $\gamma$ the sensitivity to the duration of the break is less. However, there is a loss of potency at $\gamma = 4$ when $\tau = 0.6$ relative to the split-sample algorithm. This is not surprising, because the latter is essentially optimized for that situation.

Castle et al. (2008) investigate the Autometrics block-search algorithm in the presence of multiple breaks for dynamic models, including unit-root models.

### 7 Variations of the Autometrics block search algorithm

Five variants of the block search algorithm are considered here. The difference relates to the ‘effort’ that is taken in the search for omitted variables (i.e. the expansion step):

0. is the quick version, where diagnostic tracking is always switched off during the expansion step.

1. uses the standard version of the algorithm.
1% nominal size

<table>
<thead>
<tr>
<th></th>
<th>γ = 3</th>
<th></th>
<th>γ = 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>τ = 0.6</td>
<td>0.7 0.8 0.9</td>
<td>τ = 0.6 0.7</td>
<td>0.8 0.9</td>
<td></td>
</tr>
</tbody>
</table>

3. Split-sample algorithm using Autometrics*

<table>
<thead>
<tr>
<th></th>
<th>Gauge %</th>
<th>Potency %</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ = 0.6</td>
<td>0.7 0.8 0.9</td>
<td>87.6 84.9 86.6 90.2</td>
</tr>
</tbody>
</table>

3. Autometrics block search algorithm

<table>
<thead>
<tr>
<th></th>
<th>Gauge %</th>
<th>Potency %</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ = 0.6</td>
<td>0.7 0.8 0.9</td>
<td>75.1 81.0 86.3 91.4</td>
</tr>
</tbody>
</table>

* diagnostic tracking switched off in blocks

Table 8: Experiment DGP:B with breaks of size γ after 0.6T, 0.7T, 0.8T, 0.9T. T = 100 observations, M = 1000 replications, \( p_{\alpha} = 0.01 \).

2. uses random ordering of the variables in the expansion step\(^6\) for stages B and D.

3. uses extended search: the cross-block search is used in stage C (so only once), but with diagnostic tracking switched off. For this purpose the block size is halved.

4. implements quick cross blocks: cross-block search is used in stages C and D, but with diagnostic tracking switched off. The block size is halved for the cross blocks.

5. implements standard cross blocks: cross-block search is used in stages C and D. The block size is halved. Diagnostic tracking is switched on in this mode.

Table 9 summarizes the differences between the modes of operating the algorithm. Figure 5 compares the outcomes for experiments DGP:B and DGP:S. First, comparing effort 0 to the standard mode (labelled ‘1’), we see that there is little benefit of diagnostic tracking here. This is not so surprising, because all the experiments are well behaved, once sufficient dummies have been selected. Next, randomization of blocks (effort 2) does not help with potency, but slightly deteriorates the gauge. There is some benefit in inserting a single cross-block search (effort 3) in experiment DGP:B, particularly when \( \gamma = 4 \). This is quite surprising, because the dummy variables are all orthogonal. The extra effort of further cross-block searches (‘4’ and ‘5’) does not particularly help, particularly considering that they are about twice as slow as effort 3.

So, it appears that inserting a single cross-block search could be a useful alternative to the default settings.

---

\(^6\)Resetting to initial seed at the start of the algorithm, to ensure that repeated runs of the algorithm give the same results.
Table 9: Overview of the differences between variants of the block search algorithm. Effort 1 is the default.

<table>
<thead>
<tr>
<th>Effort:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random block ordering</td>
<td>—</td>
<td>—</td>
<td>B,D</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Using cross blocks</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>C</td>
<td>C,D</td>
<td>B–D</td>
</tr>
</tbody>
</table>

Figure 5: Experiments DGP:B (circles) and DGP:S (squares) using different variants of the block search algorithm. $T = 100$, $M = 1000$, $p_a = 0.01$, $\gamma = 4$ (left panel), $\gamma = 4$ (right panel, different potency scale).
8 Indicator saturation as a robust estimator

8.1 Efficiency

Johansen and Nielsen (2009) describe a split-sample estimator for the indicator-saturated (IS) regression model and show that it is a one-step M-estimator that is iterated twice. They derive the asymptotic distribution of the first and second iteration under the null of no outliers allowing for stochastic regressors. After the first iteration, indicator saturation has the same asymptotic distribution as the ‘robustified’ least-squares estimator (i.e. OLS followed by removal of outliers), and continued iteration may lead to that of least trimmed squares (LTS) estimator of Rousseeuw, see Rousseeuw and Leroy (1987, p.135,181).

The indicator saturation estimator of \( y_t = x_t' \beta + \sum_{t=1}^{T} I_i \) is defined by Johansen and Nielsen (2009) as (assuming a half-way split):

1. Split the sample in half and saturate each half in turn with dummies, collecting the significant ones. Equivalently, Run OLS on the first half of the data \( T_1 \) to give \( \hat{\beta}_1, \hat{\sigma}_1^2 \), then on the second half \( T_2 \) to give \( \hat{\beta}_2, \hat{\sigma}_2^2 \): 

\[
\hat{\beta}_j = \left\{ \sum_{t \in T_j} x_t x_t' \right\}^{-1} \sum_{t \in T_j} x_t y_t, \\
\hat{\sigma}_j^2 = \left\{ T_j - k \right\}^{-1} \sum_{t \in T_j} (y_t - x_t' \hat{\beta}_j)^2.
\]

Create t-ratios

\[
v_t = I[t \in T_2](y_t - x_t' \hat{\beta}_1)/\hat{\sigma}_1 + I[t \in T_1](y_t - x_t' \hat{\beta}_2)/\hat{\sigma}_2,
\]

where \( I[\cdot] \) is the indicator function.

2. Run OLS on the subset corresponding to \( I[|v_t| \leq c] \) using a bias correction for the variance:

\[
\hat{\beta} = \left\{ \sum_{t=1}^{T} x_t x_t' I[|v_t| \leq c] \right\}^{-1} \sum_{t=1}^{T} x_t y_t I[|v_t| \leq c], \\
\hat{\sigma}^2 = \frac{1 - \alpha}{\tau_2^2} \left\{ \sum_{t=1}^{T} I[|v_t| \leq c] - k \right\}^{-1} \sum_{t=1}^{T} (y_t - x_t' \hat{\beta})^2 I[|v_t| \leq c]. \tag{5}
\]

Where \( \phi \) denotes the normal density, \( \Phi^{-1} \) are the normal quantiles, \( c = \Phi^{-1}(1 - \alpha/2), \tau_2^2 = 1 - \alpha - 2c\phi(c) \).

3. Construct new t-ratios for testing:

\[
\tilde{v}_t = (y_t - x_t' \hat{\beta})/\hat{\sigma},
\]

and select observations according to \( I[|\tilde{v}_t| \leq c] \).

4. Finally, run standard OLS on the subset of observations (retaining the bias correction):

\[
\tilde{\beta} = \left\{ \sum_{t=1}^{T} x_t x_t' I[|\tilde{v}_t| \leq c] \right\}^{-1} \sum_{t=1}^{T} x_t y_t I[|\tilde{v}_t| \leq c], \\
\tilde{\sigma}^2 = \frac{1 - \alpha}{\tau_2^2} \left\{ \sum_{t=1}^{T} I[|\tilde{v}_t| \leq c] - k \right\}^{-1} \sum_{t=1}^{T} (y_t - x_t' \tilde{\beta})^2 I[|\tilde{v}_t| \leq c]. \tag{6}
\]
Figure 6: Efficiency of $\beta$ for three robust estimators: first iteration of indicator saturation (IS-1step), second iteration of indicator saturation (IS-2step), LTS. Left panel: against critical value $c$; right panel: against $\alpha$, with $c = \Phi^{-1}(1 - \alpha/2)$.

The robust literature is primarily concerned with two properties of robust regression estimators. The first is the breakdown point, which is asymptotically defined, or the finite-sample breakdown point (FBP), which, loosely speaking, is the percentage of observations that can be replaced with outliers while having a limited effect on the bias. The second is efficiency: how close is the estimator variance to the maximum likelihood estimator under normality. Ideally, we have high breakdown (up to 50%), which can be preselected, and high efficiency. Figure 6 shows the efficiency of indicator saturation and LTS.

OLS has a finite-sample breakdown point of $1/T$, because just one outlying observation can cause it to break down. Least absolute deviations (LAD) is no better. M-estimation for regression solves

$$\hat{\beta}_M = \arg\min_{\beta} \sum_{t=1}^{T} \rho \left( \frac{r_t(\beta)}{\hat{\sigma}} \right),$$

where $r_t(\beta) = y_t - x_t'\beta$. Alternatively, after taking derivatives, $\psi(z) = \partial \rho(z)/\partial z$, the following non-linear system can be solved:

$$\sum_{t=1}^{T} \psi \left( \frac{r_t(\hat{\beta}_M)}{\hat{\sigma}} \right) x_t = 0.$$

Several suggestions for the $\psi$ function have been made, in particular Huber’s function, which is monotone, and Tukey’s bisquare, which is redescending. M-estimation works well when the regressors are fixed, and only outliers in the observations for the dependent variable occur. In that case, Maronna, Martin and Yohai (2006, p.101) recommend using LAD to estimate the preliminary scale $\hat{\sigma}$, followed by M-estimation using the bisquare function. This approach does not have good breakdown properties when the regressors are stochastic and there are outliers in the regressors. Alternatives with better properties include MM and $\tau$ estimates (see Maronna et al., 2006, Ch. 5, for an overview; both of these allow for a user-defined breakdown point as well as efficiency level).
Although robustified least squares and indicator saturation are more efficient than least trimmed squares, they can be very different under the alternative. Outlier-adjusted least squares, that is first estimate by ordinary least squares, then remove observations with large residuals based on a chosen critical value, was studied by Ruppert and Carroll (1980), and, not surprisingly following the example in §3, is not robust at all.

Least trimmed squares is defined as:

$$\hat{\beta}_{LTS} = \arg\min_\beta \sum_{t=1}^H r^2_{(t)}(\beta),$$

based on the ordered residuals $|r_{(1)}| \leq |r_{(2)}| \leq \ldots \leq |r_{(T)}|$. When $H$ is close to $T/2$, LTS locates that half of the observations which has the smallest variance. In that case, the breakdown point is 50%.

LTS is a so-called S-estimator, because it minimizes a measure of scale. LTS does not allow for a prespecified efficiency, and is less efficient than MM or $\tau$ estimation. Using LTS for outlier detection improves the efficiency, while maintaining the good breakdown properties:

1. Run LTS at breakdown of 35% (say, so $H = 0.65T$) to obtain $\hat{\beta}_{LTS}$ and $\hat{\sigma}_{LTS}$ based on 65% of the observations:

$$\hat{\sigma}^2_{LTS} = \frac{1 - \alpha}{c_1 \tau_1^2} \frac{1}{H - k} \sum_{t=1}^H r^2_{(t)}(\hat{\beta}_{LTS}),$$

where $c_1 = \Phi^{-1}(1 - \alpha_{1/2})$, $\tau_1^2 = 1 - \alpha_{1} - 2c_1\phi(c_1)$, and $\alpha_{1} = 1 - H/T$.

Construct $t$-ratios for testing:

$$u_t = (y_t - x_t'\hat{\beta}_{LTS})/\hat{\sigma}_{LTS}.$$

2. Select $\alpha_2$ with corresponding critical value $c_2 = \Phi^{-1}(1 - \alpha_2/2)$ and compute $\hat{\beta}, \hat{\sigma}$ from OLS on observations selected according to $I[|u_t| \leq c_2]$, incorporating the bias correction based on $\tau_2^2 = 1 - \alpha_2 - 2c_2\phi(c_2)$. So this is (5) using $u_t, c_2$ instead of $v_t, c$.

Construct $t$-ratios for testing:

$$\tilde{u}_t = (y_t - x_t'\hat{\beta})/\hat{\sigma},$$

and select observations according to $I[|\tilde{u}_t| \leq c_2]$, again with bias correction. This is (6) using $\tilde{u}_t, c_2$ instead of $\tilde{v}_t, c$.

Suggested implementations (Rousseeuw – need references) don’t run a regression at stage two, but instead use:

$$\tilde{\sigma}^2 = \left( \sum_{t=1}^r I[|u_t| \leq c_2] - k \right)^{-1} \sum_{t=1}^r u_t^2 I[|u_t| \leq c_2],$$

followed by

$$\tilde{u}_t = (y_t - x_t'\hat{\beta}_{LTS})/\tilde{\sigma},$$

and finally OLS without bias correction. However, the results in this paper suggest that the bias corrections are essential and that re-estimation is beneficial (particularly with a moderately sized break).
LTS  least trimmed squares with breakdown 100\(\alpha\)%.
RLTS  RLTS with initial breakdown 35\% and critical value \(\Phi^{-1}(1 - \alpha/2)\).
IS-2step  Indicator saturation following Johansen and Nielsen (2009) as described above.
AUTO  Autometrics indicator saturation using reduction \(p_{\alpha} = \alpha\) and with the constant and regressor always in the model (so model selection is only over the indicators).

Table 10: Estimators used in the efficiency experiment

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS-2step</td>
<td>Indicator saturation following Johansen and Nielsen (2009)</td>
</tr>
<tr>
<td>LTS (T=100)</td>
<td>Least trimmed squares with breakdown 100(\alpha)% for sample size T=100</td>
</tr>
<tr>
<td>AUTO (T=100)</td>
<td>Autometrics indicator saturation using reduction (p_{\alpha} = \alpha)</td>
</tr>
</tbody>
</table>

Figure 7: Efficiency of \(\beta\) for four robust estimators: IS-2step, LTS, RLTS and the Autometrics algorithm. Thick lines that extend to \(\alpha = 1\) are asymptotic efficiencies. Other lines are based on simulation for \(T = 100\).

3. Finally, run OLS on the latest selection. In other words, add impulse dummies for those observations that have \(|\hat{u}_t| > c_2\).

Rousseeuw and Leroy (1987) call this reweighted least squares (the weights are 0 or 1), but I shall denote this estimator RLTS, for robust least squares based on LTS.

He and Portnoy (1992) show that RLTS converges at the same rate as the initial estimator, but the precise asymptotic distribution is not known (cf. Maronna et al., 2006, p.133)). The results in Johansen and Nielsen (2009) may be extended to derive the asymptotic distribution of RLTS, see §A.3 in the Appendix. The large efficiency gain comes from moving from \(\alpha_1\) to \(\alpha_2\) (e.g. from 0.6 to 0.975), regardless of the precise asymptotic distribution.

I use the following Monte Carlo experiment to study the efficiency of the three methods sofar. The DGP is (3), extended with one regressor and without break:

\[
\text{DGP:} Bx_0 \quad y_{2,t} = x_t + u_t, \quad u_t \sim N(0,1), \quad x_t \sim N(0,1).
\]  

The model is \(y_t = \alpha + \beta x_t + \varepsilon_t\) for sample size \(T = 100\), with all indicators added in the IS and Autometrics case. The estimators are listed in Table 10.

Figure 7 plots the MSE of the estimators of \(\beta\) and \(\sigma^2\), relative to that in the absence of selection and robustification. For IS-2step and LTS this is remarkably close to the asymptotic results reported.
Table 11: Estimators used in the structural break experiment

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>ordinary least squares</td>
</tr>
<tr>
<td>LAD</td>
<td>least absolute deviations</td>
</tr>
<tr>
<td>MM</td>
<td>MM-estimation with initial location estimated by LTS with breakdown 35%, then initial scale from an M-estimate using optimal weight function, followed by M-estimation of scale using optimal weight function and efficiency 0.9.</td>
</tr>
<tr>
<td>τ</td>
<td>τ-estimation using optimal weight function with breakdown 40% and efficiency 0.9.</td>
</tr>
<tr>
<td>LTS</td>
<td>least trimmed squares with breakdown 35% (efficiency 17%).</td>
</tr>
<tr>
<td>RLTS</td>
<td>RLTS with initial breakdown 35% and critical value 2.576 (1% critical value from the normal distribution, corresponding to an efficiency of almost 90%).</td>
</tr>
<tr>
<td>IS-2step</td>
<td>Indicator saturation at 1% following Johansen and Nielsen (2009) as described above (efficiency 92%).</td>
</tr>
<tr>
<td>AUTO</td>
<td>Autometrics indicator saturation using reduction ( p_a = \alpha ) and with the constant and regressor always in the model (so model selection is only over the indicators).</td>
</tr>
</tbody>
</table>

in Johansen and Nielsen (2009) for \( \beta \). No result is available for \( \sigma^2 \) from IS-2step, so only the 1-step asymptotic values are given. Autometrics is close to RLTS for small \( \alpha \), but then diverges.

All estimators of \( \beta \) are unbiased, all except Autometrics are unbiased for \( \sigma^2 \). Although LTS shows a small bias for \( \sigma^2 \) for large \( \alpha \), and RLTS for \( \alpha \) around 0.3, but in both cases this appears to be a small sample issue. Note that the RLTS MSE for \( \sigma \) would be much worse without the proposed bias correction in the second and third stage. The autometrics algorithm has no bias correction.

### 8.2 Power against structural break

The first power results consider the ability to detect a structural break, one of the main reasons for dynamic econometric models to break down. The DGP is (7), but now extended with a break:

\[
\text{DGP:Bx } y_{3,t} = x_t + \gamma (I_{T+1} \ldots + I_T) + u_t, \quad u_t \sim \mathcal{N}(0,1), x_t \sim \mathcal{N}(0,1). \quad (8)
\]

Setting \( T = 100 \) and \( \tau = 0.7 \) so that thirty percent of the sample is in the break period. When \( \gamma = 0 \), the experiments are under the null. Note that the regressor is redrawn every time.

The initial model consists of the regressor and intercept, as well as all dummies where required:

\[
\text{MOD:Bx } y_{3,t} = \alpha + \beta x_t + \sum_{i=1}^{T} \beta_i I_i + \varepsilon_t. \quad (9)
\]

Table 11 lists the estimators that are considered. Many of the estimators are computationally demanding. LAD is based on Barrodale and Roberts (1973, 1974). All high breakdown estimators
require that a local optimum is found that is close to the global optimum, for a very non-smooth objective function. The approach is to randomly compute parameter estimates from a subsample, follow these for a few iterations, and then keep the best 10 for full iteration. Therefore, the optimization is stochastic, and there is no guarantee that the optimum can be found (although the practical performance seems to be good). For LTS I follow Rousseeuw and van Driessen (2006) with some minor modifications. For $\tau$-estimation I follow Salibian-Barrera, Willems and Zamar (2008), again with some improvements for speed and reliability.

Figure 8 plots the RMSE for the estimators listed in Table 11, except for AUTO. It is clear that OLS and LAD perform badly, with LAD getting stuck as the magnitude of the break grows. Both the MM and $\tau$-estimates have a large RMSE in the constant and variance, but recover when the break is about 6 to 7 standard errors. RLTS performs considerably better, while the inefficiency of LTS under the null (i.e. break of size zero) is clearly visible. Of course, indicator saturation would break down entirely of the data were randomized — it is essentially optimized for a break in one half of the sample.

Figure 9 also plots the RMSE, now focussing on indicator saturation, RLTS, and the Autometrics algorithm of §5. This shows the last doing well, both on the intercept and on $\sigma^2$, despite the lack of bias correction.
Figure 9: RMSE for coefficient estimates in experiment DGP:Bx. \( T = 100, M = 10000, \tau = 0.7, \alpha = 1\%, \gamma \)
on horizontal axis.

8.3 ‘Exact fit’ property

Robust estimators such as LTS, RLTS, \( \tau \) and MM (but not M) have the so-called exact fit property. This implies that, if there is a subset of the data that follows a different model, this will be ignored by the robust estimator. This subset needs to be a smaller proportion of the data than the breakdown point. Figure 10 illustrates this with one sample from the following DGP with \( k = 1 \):

\[
\begin{align*}
x_{i,t} &= v_t, \quad v_t \sim \text{IN}(0, 1), \quad t \in T_1, \\
y_{4,t} &= \sum_{i=1}^{k} x_{i,t} + u_t, \quad u_t \sim \text{N}(0, 1), \quad t \in T_1, \\
x_{i,t} &= v_t, \quad v_t \sim \text{IN}(50, 25), \quad t \in T_2, \\
y_{4,t} &= u_t, \quad u_t \sim \text{N}(0, 25). \quad t \in T_2.
\end{align*}
\]

(10)

The model for estimation is given by

\[
y_{4,t} = \beta_0 + \sum_{i=1}^{k} \beta_i x_{i,t} + \varepsilon_t, \quad t = 1, \ldots, T.
\]

This is augmented with the \( T \) indicators when relevant.

As Figure 10 shows, OLS would attempt to fit all data points. LTS, on the other hand, ignores the subset of the model that belongs to \( T_2 \), which is 30\% of the sample here. Table 12 shows the result for DGP (10) with 10 regressors. RLTS is able to detect the subset of observations in \( T_2 \), whether they are located at the end or randomly distributed. However, indicator saturation and the Autometrics version of that can only detect the second group if it is a consecutive block, but not if it is randomly distributed.
Figure 10: Exact fit property of LTS

<table>
<thead>
<tr>
<th>β₁, ..., β₁₀</th>
<th>Average RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1 = {1, ..., 90}$</td>
<td>$T_1$ random, dim $T_1 = 90$</td>
</tr>
<tr>
<td>$T_2 = {91, ..., 100}$</td>
<td>$T_2$ random, dim $T_2 = 10$</td>
</tr>
<tr>
<td>RLTS</td>
<td>0.126</td>
</tr>
<tr>
<td>IS-2step</td>
<td>0.119</td>
</tr>
<tr>
<td>AUTO</td>
<td>0.118</td>
</tr>
</tbody>
</table>

Table 12: Illustration of exact fit property, DGP (10), $k = 10$, $T = 100$, $M = 1000$, 10% of sample in second ‘regime’.

8.4 Structural breaks in larger models

In practice, initial models may have several or many irrelevant variables, and automatic selection procedures help in finding good models in such settings. Therefore, we consider a more realistic data generation process, which allows for irrelevant variables:

$$
y_t = 0.1 + \sum_{i=1}^{k_s} \alpha_j x_{i,t} + \beta_0 y_{t-1} + \sum_{i=1}^{k_s} \beta_j x_{i,t-1} + 6I[t > 0.8T] + u_t, \quad u_t \sim \text{IN}[0, 1],$$

$$
x_{i,t} = v_{i,t}, \quad i = 1, ..., k + m, \quad v_{i,t} \sim \text{IN}[0, 1],$$

$$
t = -p, ..., 1, ..., T.$$

(11)
regressors. The final results are then based on OLS on the model (12) augmented with the selected indicators (and without any bias correction for indicators). For the other two, IS-twostep(*) and RLTS(*), the initial estimated model consists of the DGP variables augmented with dummies only. The next plot gives the average RMSE, computed over the 10 relevant variables, $k^*$ for $m = 10$.

Three estimation procedures are compared. The first is Autometrics (AUTO) with the proposed algorithm for more variables than observations. In that case, the selection is over all $x$-variables, all indicator variables, and, in the dynamic experiment, the lagged dependent variable. The intercept is always included. For the other two, IS-twostep(*) and RLTS(*), the initial estimated model corresponds to (12) without the indicators. Then, indicator saturation and RLTS is used to select the indicators. The final results are then based on OLS on the model (12) augmented with the selected indicators (and without any bias correction for $\hat{\sigma}^2$). Both of these are only defined for fewer than $T/2$ relevant variables.

Figures 11 and 12 report the results for the static and dynamic experiments respectively. The first plot, labelled ‘Dummies (relevant)’, reports the percentage of the correct dummies included in the model on average (for $M = 1000$ replications). This is the potency computed over the indicator dummies only. The next plot gives the average RMSE, computed over the 10 relevant $x$-variables only.
Figure 11: Model selection and robust estimation for a static regression in the presence of a structural break and irrelevant variables.

Figure 12: Model selection and robust estimation for a dynamic regression in the presence of a structural break and irrelevant variables.

\((x_{1,t}, \ldots, x_{10,t} \text{ in the static case, } x_{1,t}, x_{1,t-1}, \ldots, x_{5,t-1} \text{ in the dynamic case})\). The third plot reports the average RMSE over the irrelevant \(x\)-variables. On the horizontal axis of all graphs is the total number of irrelevant \(x\) variables, ranging from 0 to 90 in the experiments (see Table 13). The final plot gives the average estimated \( \hat{\sigma}^2 \).

We see in Figure 11 that indicator saturation and RLTS are more successful when there are no
irrelevant variables. The RMSE over relevant variables is almost twice for Autometrics, although half of the others for the irrelevant variables. Autometrics also underfits somewhat. However, as the number of irrelevant variables grows, IS and RLTS steadily deteriorate, with the RMSE getting worse than AUTO when the number of irrelevant variables exceeds fifteen. It is remarkable how steady the performance of the Autometrics algorithm is in these experiments.

Some of these results are accentuated in the dynamic case, Figure 12. Except for IS, it is much harder to detect the correct dummies. Nonetheless, Autometrics has better RMSE when there are 15 or more irrelevant variables. Now both IS and RLTS start overfitting heavily as the number of irrelevant variables increases, which implies that they select an increasingly larger amount of irrelevant indicators.

*** NB. These results do not use bias correction for Autometrics (not unclear how well it works for $T > k$). However, a bias correction for $\hat{\sigma}^2$ may perhaps be useful.

9 Mis-specification testing and robust estimation

10 Conclusion

Acknowledgements

This research was supported by ESRC grant RES-062-23-0061. Helpful comments and suggestions from David Hendry are gratefully acknowledged.
A Appendix

A.1 Asymptotic distribution of variance of IS-2step

JN refers to Johansen and Nielsen (2009). Assuming symmetric f, and starting with equation (JN-1.21) with second step variance \( \tilde{\sigma}^2_{x,t} = \tilde{\sigma}^2_{(2)} \):

\[
\tau_c^s T^{\frac{1}{2}} \left( \tilde{\sigma}^2_{(2)} - \sigma^2 \right) = T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 - \sigma^2 b_s) w_t + \frac{1}{2} \zeta^{c_3} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 - \sigma^2) h_t + o_p(1),
\]

(13)

where \( b_s = \tau_c^s / (1 - \alpha) \) is the bias correction for the variance and \( w_t = I(\varepsilon_t < \sigma) \) the 0, 1 weights. Analogous to the proof of Theorem JN-1.7, use expression (JN-1.25), which is the estimator of the variance in the first step:

\[
\tau_c^s T^{\frac{1}{2}} \left( \tilde{\sigma}^2_{(1)} - \sigma^2 \right) = T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 / \sigma - \sigma b_s) w_t + \frac{1}{4} \zeta^{c_3} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 / \sigma - \sigma) h_t + o_p(1).
\]

Insert

\[
\tau_c^s T^{\frac{1}{2}} \left( \tilde{\sigma}^2_{(1)} - \sigma^2 \right) = \frac{1}{2} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 / \sigma - \sigma b_s) w_t + \frac{1}{4} \zeta^{c_3} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 / \sigma - \sigma) h_t + o_p(1)
\]

into (13):

\[
\tau_c^s T^{\frac{1}{2}} \left( \tilde{\sigma}^2_{(2)} - \sigma^2 \right) \\
= T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 - \sigma^2 b_s) w_t + \frac{1}{2} \zeta^{c_3} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 - \sigma^2) h_t + o_p(1)
\]

\[
+ \frac{1}{4} \zeta^{c_3} T^{-\frac{1}{2}} \sum_{t=1}^{T} (\varepsilon_t^2 - \sigma^2) h_t + o_p(1)
\]

(14)

where \( g = \frac{1}{2} \zeta^{c_3} / \tau_c^s \). Computing the conditional variance of the variance as in JN, this converges in probability to \( \sigma^4 \pi_{(2)} \):

\[
\pi_{(2)} = (\tau_4^s + b_s^2 (1 - \alpha) - 2 \tau_2^s b_s) (1 + g) + \frac{1}{4} (\zeta_3^s)^2 (\tau_4 - 1) g^2
\]

\[
+ \zeta_3^s (\tau_4^s + b_s (1 - \alpha) - \tau_2^s - b_s \tau_2^s) (1 + g) g
\]

\[
+ \frac{1}{4} (\zeta_3^s)^2 (\tau_4 - 1) g^2 + \zeta_3^s (\tau_4^s - b_s \tau_2^s) (1 + g) g.
\]

So the efficiency of the variance of the two-step IS estimator is \( \eta_{\sigma_{(2)}^{-1}} = 2(\tau_c^s)^2 / \pi_{(2)} \).
A.2 Iterated IS

Again assume symmetry and follow the proof of Theorem JN-1.7. Using equation (JN-1.20) with \( N_T = T^{-1/2} I_T \):

\[
(1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_{(2)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t w_t + \frac{\xi^c}{1 - \alpha} (1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_{(1)} - \beta \right) + o_p(1). \tag{15}
\]

Insert (JN-1.24), which is:

\[
(1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_{(1)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t w_t + \xi^c T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t h_t + o_p(1), \tag{16}
\]

into (15):

\[
(1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_{(2)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t w_t + \frac{\xi^c}{1 - \alpha} \xi^c T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t h_t + o_p(1).
\]

So for the \( n \)-step estimator:

\[
(1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_{(n)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t \left\{ w_t \left[ \sum_{j=0}^{n} \left( \frac{\xi^c}{1 - \alpha} \right)^j \right] + \left( \frac{\xi^c}{1 - \alpha} \right)^n \xi^c h_t \right\} + o_p(1). \tag{17}
\]

If we let \( n \to \infty \), because \( \frac{\xi^c}{1 - \alpha} < 1 \), I conjecture that (this requires the \( o_p(1) \) term not to blow up):

\[
(1 - \alpha) \Sigma^{1/2} \left( \hat{\beta}_n - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t \left( \frac{1 - \alpha}{1 - \alpha - \xi^c} \right) + o_p(1). \tag{18}
\]

The conditional variance converges in probability towards:

\[
\sigma^2 \Sigma \left[ \frac{\tau^2}{(1 - \alpha - \xi^c)^2} \right].
\]

Then

\[
T^{1/2} \left( \hat{\beta}_n - \beta \right) \overset{D}{\sim} N_k \left[ 0, \sigma^2 \Sigma^{-1} \frac{\tau^2}{(1 - \alpha - \xi^c)^2} \right]. \tag{19}
\]

Under normality \( \tau^2 = 1 - \alpha - \xi^c \), in which case the efficiency is \( 1 - \alpha - \xi^c \). Iterating one more time from \( \hat{\beta}_n \) does not change the asymptotic distribution. The distribution of \( \hat{\beta}_n \) is the same as that of LTS.

A.3 Two different \( \alpha \)'s

Start from the \( n \)-step estimator (17) using \( p \)-value \( \alpha_1 \) with corresponding critical value \( c_1 \):

\[
(1 - \alpha_1) \Sigma^{1/2} \left( \hat{\beta}_{(n)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t w_t \left[ \sum_{j=0}^{n} \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^j \right] + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^n \xi_j^c T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t h_t + o_p(1).
\]

\[
\sum_{j=0}^{n} \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^j = \frac{(1 - \alpha_1)^{n+1}}{n+1} + \frac{\xi_j^c}{1 - \alpha_1} + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^2 + \cdots + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^{n-1}.
\]

Asymptotically

\[
\left( \frac{\xi_j^c}{1 - \alpha_1} \right)^n \xi_j^c = \sum_{j=0}^{n} \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^j + \cdots + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^{n-1}.
\]

\[
\sum_{j=0}^{n} \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^j = \frac{(1 - \alpha_1)^{n+1}}{n+1} + \frac{\xi_j^c}{1 - \alpha_1} + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^2 + \cdots + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^{n-2}.
\]

\[
\sum_{j=0}^{n} \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^j = \frac{(1 - \alpha_1)^{n+1}}{n+1} + \frac{\xi_j^c}{1 - \alpha_1} + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^2 + \cdots + \left( \frac{\xi_j^c}{1 - \alpha_1} \right)^{n-1}.
\]
where \( w_t = I[|\varepsilon_t| < c_1 \sigma] \). Then take the next step using \( p \)-value \( \alpha_2 \) with corresponding critical value \( c_2 \):

\[
(1 - \alpha_2) \Sigma T^{1/2} \left( \tilde{\beta}_{(n+1)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t u_t + \xi^c_1 \frac{\xi^c_1}{1 - \alpha_1 - \xi^c_1} + o_p(1),
\]

where \( u_t = I[|\varepsilon_t| < c_2 \sigma] \). Substitution gives:

\[
T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t \left\{ w_t + w_t \xi^c_1 \frac{\xi^c_1}{1 - \alpha_1} \sum_{j=0}^{n} \left( \frac{\xi^c_1}{1 - \alpha_1} \right)^j \right\} + \xi^c_1 + o_p(1).
\]

Note that \( u_t w_t = u_t \) when \( \alpha_1 > \alpha_2 \) and \( u_t w_t = w_t \) otherwise.

Letting \( n \to \infty \), because \( \xi^c_1 / (1 - \alpha_1) < 1 \) I conjecture (but note that we could have started directly from (18):

\[
(1 - \alpha_2) \Sigma T^{1/2} \left( \tilde{\beta}_{(n+1)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t \left\{ u_t + w_t \frac{\xi^c_2}{1 - \alpha_1 - \xi^c_1} \right\} + o_p(1).
\]

When \( \alpha_1 > \alpha_2 \), the conditional variance converges in probability towards:

\[
\sigma^2 \Sigma \left[ \tau^2 + \tau^2 \xi^c_1 \left( \frac{\xi^c_2}{1 - \alpha_1 - \xi^c_1} \right)^2 + \tau^2 \frac{\xi^c_2}{1 - \alpha_1 - \xi^c_1} \right].
\]

One additional step:

\[
(1 - \alpha_2) \Sigma T^{1/2} \left( \tilde{\beta}_{(n+2)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t \left\{ u_t \left( 1 + \frac{\xi^c_2}{1 - \alpha_2} \right) + w_t \frac{\xi^c_2}{1 - \alpha_2} - \frac{\xi^c_2}{1 - \alpha_1 - \xi^c_1} \right\} + o_p(1).
\]

Then, writing

\[
q_1 = \frac{\xi^c_2}{1 - \alpha_1 - \xi^c_1}, \quad q_2 = \frac{\xi^c_2}{1 - \alpha_2},
\]

we find:

\[
T^{1/2} \left( \tilde{\beta}_{(n+2)} - \beta \right) \overset{D}{\sim} \mathcal{N}_k \left[ 0, \sigma^2 \Sigma^{-1} \frac{q_1^2 \tau^2 + q_2^2 \tau^2 + 2q_1 q_2 \tau^2 \xi^c_1}{(1 - \alpha_2)^2} \right]. \tag{20}
\]

Figure 13 shows the efficiency of the one and two-step continuations from an initial breakdown point of \( \alpha_1 = 0.2, 0.35, 0.5 \). The lines are bounded by the efficiency of \( \tilde{\beta}_k \). The right panel of Figure 13 zooms in on the interesting part of the left panel. For example, a breakdown of \( \alpha_1 = 35\% \) followed by the two-step estimator at \( \alpha_2 = 2.5\% \) has an efficiency of close to 75\%. Selecting \( \alpha_2 = 1\% \) results in an efficiency of almost 90\%. The efficiency of the variance could be derived in a similar fashion.

Iterating fully removes the effect of \( \alpha_1 \):

\[
(1 - \alpha_2) \Sigma T^{1/2} \left( \tilde{\beta}_{(n+)} - \beta \right) = T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t u_t \left( 1 - \frac{\alpha_2}{1 - \alpha_2 - \xi^c_1} \right) + o_p(1).
\]
Efficiency of $\beta_{*+1}$ and $\beta_{*+2}$

Figure 13: Efficiency of $\tilde{\beta}_{*+2}$ and $\tilde{\beta}_{*+1}$ for three choices of $\alpha_1$; $\alpha_2$ is on the horizontal axis.

References


