Review of PcGets 1 for Windows

GUNNAR BÅRDSSEN

Department of Economics, Norwegian University of Science and Technology, Dragvoll, 7491 Trondheim, Norway
E-mail: gunnar.bardsen@svt.ntnu.no

Received: October 2001

Summary  PcGets is a computer program that implements automated econometric model selection. After discussing the installation and the documentation of the program, I describe the interface and the algorithm. Before concluding, I give an illustration of use of the program.

Keywords:  Econometric modelling, Econometric computing, Program development.

1. GENERAL OVERVIEW

PcGets implements econometric model selection from a general specification by automatic search procedures. The program can be used interactively or in batch mode, ensuring easy replication of any modelling session. PcGets uses GiveWin 2 as a shell for output and data management. Being a part of the Oxmetrics suite of programs that use GiveWin as output–input central, the program interacts seamlessly with the latest version of PcGive. PcGets is installed in GiveWin as a so-called Ox-pack—compiled code written in Ox, see Doornik (2001).

The program is designed to analyse linear models. The estimation routines comprise OLS, instrumental variables, recursive estimation—both forward and backwards—as well as rolling regressions. Post-estimation evaluation includes tests of misspecification, collinearity analysis, graphic analysis, dynamic analysis, forecasting, and specification tests.

In addition to the programs PcGets and GiveWin, the package consists of three data sets, three batch files, an impressive help-system in html-format and the GiveWin documentation, contained in Doornik and Hendry (2001a). Finally, there is the program manual (Hendry and Krolzig (2001)), written by the authors of the program. The price of a single user licence is £250.

The documentation gives no indication of where to get technical support. However, the program is distributed by Timberlake Consultants Ltd (http://www.timberlake.co.uk), but no information about PcGets was available online at the time of writing. The program was tested running under Windows 2000.

2. INSTALLATION

PcGets is supplied on a CD-ROM, which also contains the files for the other modules in the Oxmetrics suite. The CD is equipped with a nice auto-run menu, so finding the programs is hassle-free. You have to install GiveWin 2 before you install PcGets. The programs need separate
registration codes. You need to supply the registration code both during installation and when running the program for the first time. Installing the programs under Windows 2000 did not reveal any problems.

3. DOCUMENTATION

The documentation consists of a comprehensive help system in html-format, and the manual Hendry and Krolzig (2001). The former is available online at http://www.pcgive.com/pcgets/. Hendry and Krolzig (2001) is a book of 232 pages. It is written in the same style that users of PcGive have come to expect: a user manual that also serves as a more or less complete textbook covering the econometric theory and methodology behind the program.

The first part of the book consists of an overview, and an introduction to loading and manipulating data in Chapter 2. The tutorials in the second part are the most important component of the book for most practitioners, while part three contains the theoretical background for the design of the program. Parts four and five are references pertaining to the statistics and the calibration of the algorithm.

With a new concept like this, the tutorials are extremely important. The topics covered are Model Formulation and Estimation, Post-Estimation Model Evaluation, Automatic Model Selection, Cross-section Model selection, Batch Language, and Modelling VARs. In general the tutorials are focussed and informative with lots of screen shots, but they are all on the brief side. Even though the key chapter Automatic Model Selection is by far the longest with 23 pages, more—and messier—‘real-life’ examples are called for.

More length could also be desired in the really intriguing bits: Chapters 10 and 11 in part 3 of the book, which cover The Econometrics of Model Selection and Refuting Potential Criticisms of PcGets. First the authors describe the methodology of the program, which is clearly controversial, but with very convincing evidence presented. The potential criticism is then rebutted in Chapter 11. I do doubt, however, that the discussion offered is enough to win any skeptics over.

In all fairness it must also be mentioned, as the authors also point out, that the econometric theory of the program is available in several other publications, notably Hendry (2000), Hendry and Krolzig (1999), Krolzig (2001), and Krolzig and Hendry (2001).

4. DESCRIPTION

The development of PcGets was stimulated by the results of Hoover and Perez (1999). They developed an algorithm to mimic a ‘general-to-specific’ modelling approach. They were then able to reverse the results of Lovell (1983) that specification search as a modelling strategy did not work. Building on their results, the PcGets program is simply a fully automated version of David Hendry’s ‘general-to-specific’ research methodology put into practice. It starts from a general, possibly dynamic, statistical model—the general unrestricted model (GUM). After estimation of the GUM, the program takes this specification as a benchmark and looks for statistically valid specifications. This is important: the final model must be hidden inside the GUM. If you have not thought hard enough about how your model is likely to be, the program cannot find it. So the program cannot compensate for a lack of economics. But if you have done your homework, PcGets can cut down on the time you spend on finding your best model. So how does it work?
PcGets is one of several modules that all use GiveWin as their central for input and output. So, undertaking an econometric specification search with PcGets starts with GiveWin2, which handles the data and reports all output, including graphics. GiveWin has its own file format, but imports standard file formats, like ascii, spreadsheets, Gauss- and Stata-, and matrix-formats. One of the strong sides of GiveWin has always been its flexible graphics, and this facility has become even better with Version 2. The interactive menu now includes nearly 50 different graphs—including several 3D varieties. The graphs can be saved in the internal format, or exported to postscript, encapsulated postscript, Windows metafile, or enhanced metafile. The interface is also improved in version 2, with a navigation menu on the left-hand side that also acts as a bookkeeping device: showing the user the currently available graphical windows, text windows or databases.

First the program finds benchmark values for all misspecification tests. These critical values are taken from the GUM. If you insist on using a general specification that has, say, non-normal residuals at the 5% significance level, the significance level for that test is lowered and the program will evaluate the different specifications according to a test for normality with lower levels of significance.

Second, it starts to throw out insignificant variables. This proceeds in stages. First it checks for joint significance of lags. If all the variables at that longest lag are jointly insignificant, they are permanently deleted. Then it checks whether all variables with $t$-tests below the highest preset significance level are jointly insignificant. If this test is accepted, the joint significance of the previous block together with all variables with $t$-values below the next level are tested. These cumulative $F$-tests are performed until a test rejects. The significance level is fairly loose—maybe 90%. The reason for this is that all variables that are found to be jointly insignificant are then permanently deleted. This reduced model is then the new GUM. The same set of tests are
then run again until rejection. Then a second $F$-test is run. This starts by including the most significant variables and cumulates until the test cannot reject.

At all stages all the diagnostic tests are run. If any of the tests rejects, that reduction is invalid and the variable is included again.

In the third stage the program does multi-path selection of specific models. What does this mean? One of the arguments against ‘general-to-specific’ is that as a data-based approach it is path specific. Had the researcher started the simplification with a different variable, the final model would/could have been different. Effectively, the program rebuts this critique by starting the simplification by deleting each different insignificant variable in turn, so that all possible valid simplification paths are tried. The search path is concluded the moment any specification test rejects.

Finally, the program compares all the models arrived at during the different path-dependent searches by encompassing. If no model encompasses the others, a final model is chosen on the basis of information criteria.

One of the really nice features about the program is the possibilities for calibration. It comes with three ready-made research ‘profiles’: liberal—to minimize possibility of accidental deletion; conservative—to minimize the possibility of accidentally keeping irrelevance; and then expert mode . . .

Another feature of particular interest when testing competing economic hypotheses is the ability to retain a variable in the model across specifications—fixing it.
5. AN ILLUSTRATION

As a means of a quick illustration of the potential pitfalls in specifying the GUM, consider the following stylized data-generating process of a cointegrated price $P$ and quantity $Q$ system:

$$\begin{align*}
\Delta P_t &= 0.5 \Delta Q_t + 0.5 \Delta S_t - 0.5 (P - Q - S)_{t-1} + u_p t \\
\Delta Q_t &= -0.5 \Delta P_t + 0.5 \Delta D_t - [Q + P - 0.5 D]_{t-1} + u_q t,
\end{align*}$$

where the common trends $S$ and $D$ are random walks, and all errors are NID. The corresponding $I(1)$ levels representation is

$$\begin{align*}
P_t &= 0.5 P_{t-1} + 0.5 Q_t + 0.5 S_t + u_p t \\
Q_t &= -0.5 P_t - 0.5 P_{t-1} + 0.5 D_t + u_q t.
\end{align*}$$

In addition, I created two more random walks $X_1$ and $X_2$ to enter as potential explanatory variables.

Concentrating on the price equation, a quick Monte Carlo with 1000 replications using PcNaive—see Doornik and Hendry (2001b)—revealed that OLS estimation of $P$ on all variables with two lags should result in insignificant $Q_t$ and significant $D_t$. PcGets in liberal mode correctly deleted the second lag, and indeed deleted the endogenous $Q_t$, keeping $D_t$ instead.

However, when correctly treating $Q_t$ as endogenous, and $D_t$ as an instrument, the correct model was retained without any problems—using the conservative strategy.

The results from the Instrumental Variables specification search are reported below, using the wonderful LATEX output feature of the program:
General, 3 - 101.

\[
\tilde{P}_t = -1.65 + 0.4 P_{t-1} + 0.0811 P_{t-2} - 0.0663 Q_{t-1} + 0.0479 Q_{t-2} + 0.529 S_t + 0.0303 S_{t-1} - 0.0418 S_{t-2} + 0.0566 D_{t-1} - 0.00382 D_{t-2} + 0.0728 x_1 t + 0.0427 x_1 t_{t-1} - 0.143 x_1 t_{t-2} - 0.144 x_2 t + 0.131 x_2 t_{t-1} - 0.0294 x_2 t_{t-2} + 0.47 Q_t
\]

Estimation statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSS</td>
<td>86.8011</td>
</tr>
<tr>
<td>(\hat{\sigma})</td>
<td>1.0289</td>
</tr>
<tr>
<td>(R^2)</td>
<td>0.8748</td>
</tr>
<tr>
<td>(\bar{R}^2)</td>
<td>0.8503</td>
</tr>
<tr>
<td>lnL</td>
<td>6.5093</td>
</tr>
<tr>
<td>AIC</td>
<td>0.2119</td>
</tr>
<tr>
<td>HQ</td>
<td>0.3922</td>
</tr>
<tr>
<td>SC</td>
<td>0.6576</td>
</tr>
<tr>
<td>T</td>
<td>99</td>
</tr>
<tr>
<td>p</td>
<td>17</td>
</tr>
<tr>
<td>(F_{pNull})</td>
<td>0.0000</td>
</tr>
<tr>
<td>(F_{p\text{Const}})</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Misspecification tests

\[\chi^2_{\text{normality}} = 0.047 [0.9766] \quad F_{ARCH(1-4)} = 0.715 [0.5844] \quad F_{hetero} = 0.594 [0.9398] \]

Specific, 3 - 101.

\[
\tilde{P}_t = + 0.499 P_{t-1} + 0.5 S_t + 0.487 Q_t
\]

Estimation statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSS</td>
<td>98.3990</td>
</tr>
<tr>
<td>(\hat{\sigma})</td>
<td>1.0124</td>
</tr>
<tr>
<td>(R^2)</td>
<td>0.8580</td>
</tr>
<tr>
<td>(\bar{R}^2)</td>
<td>0.8551</td>
</tr>
<tr>
<td>lnL</td>
<td>0.3014</td>
</tr>
<tr>
<td>AIC</td>
<td>0.0545</td>
</tr>
<tr>
<td>HQ</td>
<td>0.0863</td>
</tr>
<tr>
<td>SC</td>
<td>0.1332</td>
</tr>
<tr>
<td>T</td>
<td>99</td>
</tr>
<tr>
<td>p</td>
<td>3</td>
</tr>
<tr>
<td>(F_{pNull})</td>
<td>0.0000</td>
</tr>
<tr>
<td>(F_{p\text{GUM}})</td>
<td>0.7702</td>
</tr>
</tbody>
</table>

Diagnostics

\[\chi^2_{\text{normality}} = 0.089 [0.9564] \quad F_{ARCH(1-4)} = 1.013 [0.4049] \quad F_{hetero} = 0.793 [0.5778] \]

Long-run equilibrium.

\[
P = + 0.972 Q + 0.998 S
\]
I have to be honest and confess to branding a ‘data-mining’/’automatic model selection’ program to be an utterly bad idea when I first heard about it. The reasons for this was a fear of economic lobotomizing in the art/science of econometric model building in the name of the ‘general-to-specific’ tradition—in the sense of the quality of the models developed deteriorating and only the ‘data-mining’ remaining. I think I was wrong.

The benefits one gets out of using PcGets really depend on the motivation and needs of the user. Make no mistake: it is quite possible to produce disastrous models using this software. The worst possible client would be someone looking for a quick fix, claiming the results to be generated according to Professor Hendry’s scientific standards—and using the latest advances in modelling technology. ‘Garbage in, garbage out’ is as true as ever. But trivially, the same argument could be made for any nonsensical use of economic and econometric Ockhams razors. It’s just that this particular razor is a bit sharper than usual. In that sense PcGets is an expert tool.

However, the more one is willing to invest in economic theory, econometric theory, and econometric modelling experience, the more productive the use of the program is bound to be. In that sense PcGets not only offers increasing returns to knowledge, but it also promises to be an effective tool in debunking any scientific illusion in empirical macroeconomics—see Summers (1991).

So, after this slightly schizophrenic argument, one might ask: who could benefit from using this program? Off the top of my head I can think of the following characters, given the caveat of sensible input.

The student: working backwards from the end result, the serious student can learn an awful lot in a very short time about why a particular model emerged. Since PcGets will beat almost any expert on home ground, the program is bound to be a very valuable teacher in the art and science of model building and applied econometrics.

The economic theorist: the program offers a unique opportunity to expose testable hypotheses about one’s pet theoretical model to real data, without spending years on econometrics training in advance. The option of keeping a regressor fixed across specification searches is invaluable, as it will really demonstrate whether a variable belongs in a model, or not.

The consultant: PcGets provides a devastating tool for producing the best linear predictor given the information available. Get one, or you will be run out of business.

The empirical econometrician: this is probably the kind of user that will benefit the most. Analyses that earlier could have taken years can now be done in seconds. For these clients the introduction of the program is a jump in technical progress comparable to the advent of computers after manual calculating machines.

The Macro-econometric model maintainer: when new data arrive, remodelling can be done very easily indeed. In addition PcGets offer a unique opportunity to do sensitivity analysis across different specifications very quickly. Again the possibility of keeping the status quo, through fixing variables, offers a unique possibility for a progressive model building strategy.

The theoretical econometrician: due to the possibilities of calibration, the program can act as a research laboratory, and can thus act as a tool in enhancing knowledge about the behaviour of tests and research strategies. This effect is clearly demonstrated by the stream of new research results obtained by the authors through the development of the program. It would be desirable for Monte Carlo experiments to be made possible in future versions of the program.
ACKNOWLEDGEMENT

I want to thank Neil Shephard for helpful comments.

REFERENCES


