Review of SsfPack 2.2: statistical algorithms for models in state space

MARIUS OOMS
Econometric Institute, Erasmus University Rotterdam, 3062 PA Rotterdam,
The Netherlands
E-mail: ooms@few.eur.nl

Received: January 1999

Summary  SsfPack is a set of procedures with a strong link to the object-oriented matrix programming language Ox. In this review I first discuss installation issues and prerequisites for a productive use of SsfPack, namely statistical and econometric knowledge, programming skills and additional software. Next I discuss the documentation and sample programs. Before concluding I discuss extensions in the direction of user-friendly programs.

Keywords:  Econometric computing. Econometric programming. Program development.

1. WHAT IS SSFPACK 2.2?

SsfPack 2.2 consists of a number of computer files which can literally be downloaded in a minute. There are two files with canned functions, sffpack.dll and sffpack.h, 18 sample programs in Ox 2.1, and three sample data sets. Finally, there is the documentation in Koopman, Shephard and Doornik (1999), further abbreviated as KSD. The appendix of KSD gives a detailed overview of the available functions and programs.

The core of the package consists of a dynamic link library (.dll) file and a corresponding Ox header (.h) file. These files contain four basic functions, written in C, that allow for Multivariate Diffuse Kalman Filtering, Smoothing, and repeated Simulation Smoothing for Gaussian State Space models. In principle one should be able to use these functions in other programming languages as well, but the focus of the remaining part of SsfPack 2.2 is on the use in the Object Oriented Matrix Language Ox, version 2.0 or later. To this end, the dynamic link library is extended with other functions that let the basic functions come to life in Ox. I tested SsfPack with Ox 2.0 under Windows 95 and under Windows NT.

2. PREREQUISITES

For a fruitful use of SsfPack one has to invest in acquiring the appropriate econometric knowledge, programming skills, computing power and additional software. What is easy and efficient to the authors, may not be so to other potential users of SsfPack.

The basic statistical and econometric background is provided by Harvey (1993, Chs 3, 4) who describes Kalman filtering and smoothing, setting up structural models, and putting regression
models and autoregressive moving average (ARMA) models in state space form. *SsfPack* now provides the means to get to grips with these basics as well. The more advanced econometrics is not available in textbook form, but Harvey (1989) provides the background to understand the ‘nonparametric’ continuous time example and the multivariate notation used in the documentation. The exact bootstrap test against a unit root and the Bayesian estimation presented in Sections 6.4 and 6.5 are quite advanced examples of simulation-based inference, which probably are not so easy to adapt and extend yet.

*SsfPack* is not a menu-driven package and cannot easily be made into one. Basic but adequate programming skills are required to obtain results. In order to benefit from the examples in KSD one should invest some time to get used to the C-like syntax of Ox. Listing 1 in KSD provides the first example. Inexperienced programmers should note that function headers are contained in a separate .h file, curly braces mark begins and ends of blocks of statements, ampersands are used to return values of a function in an argument. Another trivial, but important aspect is that counting starts at 0, see Listing 6.

A consistent programming style is important for the readability and maintainance of programs. The so-called Hungarian notation, i.e. a decoration of variable names with prefixes, helps to indicate the scope and type of the different variables. This is clearly shown in the first part of Listing 9. The prefix *s* indicates the variable is static (i.e. hidden from other source files) and the following character indicates type, i.e. m(atrix), c(ount), v(ector), d(ouble). The prefix *p* in the declaration of *ArmaLogLik()* indicates a pointer to a function argument. Again, the learning of a consistent style requires some investment, but it is worth the effort, especially if the programs are part of longer-lasting research projects.

Econometricians who are accustomed to programming in matrix languages such as GAUSS or MATLAB, should be able to grasp Ox’s efficient preprogrammed matrix functions, statistical procedures and optimization routines quite easily. All in all one should count on 4 to 5 days to acquire the specific Ox programming skills, which would enable one to adjust the example programs of *SsfPack* to one’s specific needs without any trouble. Doornik, Draisma and Ooms (1998) is a 90 page booklet that provide the ‘hands-on’ material to obtain these skills.

One needs a fast personal computer (1998/1999 standard or later) to benefit fully from the advanced features of *SsfPack* in serious applications. The authors state that most functions are computationally efficient, but efficiency is after all a relative concept. Simulation-based inference is basically feasible, i.e. one can expect reasonably precise answers within minutes, rather than within hours or days, for the simple models. This requires, in particular, a small dimension of the state vector, say \(m < 10\), and a moderate number of observations, \(n < 1000\), see also Section 7 of KSD. The univariate local level model which is used throughout KSD in the example programs is simple enough: adequate results are obtained in seconds.

Next to Ox, with a command-line version free for academic use, one should preferably have a graphical ‘outlet’ to interpret the outcomes easily. The programs OxRun and GiveWin 1.2 (Doornik and Hendry1999) provide a suitable environment to view and manipulate the graphs under recent Microsoft Windows operating systems, as the graphs in KSD show. In contrast to *SsfPack* and Ox, GiveWin is not available for Unix platforms. Unix users can view the graphs produced by Ox, e.g. in encapsulated postscript format, but they cannot manipulate the graphs easily.

The multi-purpose editor OxEdit, which is available as a shareware product from [www.oxedit.com](http://www.oxedit.com) provides a very useful fully Microsoft Windows compatible programming environment for the development of Ox programs. It provides syntax highlighting, menus for program project management, running, linking and parsing of programs and output. Finally, a modern
SsfPack review

163

browser such as Internet Explorer or Netscape should be available to view the documentation of Ox. In sum, one should invest quite a bit to benefit from SsfPack. If one only wants to produce forecasts and forecast intervals for simple time series models, one is better off buying and using a menu-driven package. However, if one really wants to have control over the algorithms in order to experiment with more realistic, and therefore somewhat more involved models, or with modern methods of inference it is definitely worth the investment. Moreover, it provides a relatively easy way to check results from badly documented menu-driven packages. The authors have clearly demonstrated the wide range of applicability.

3. INSTALLATION

Fortunately, installation of programs in the Windows 95/98 or Windows NT environment has become a simple issue. If one has control over the folders concerned, the menu-driven installation of Ox and the accompanying programs is trivial. In order to use SsfPack one simply has to create an SsfPack folder and put four files into it, probably adjusting the names so as to make them compatible with the operating system one uses. The last steps of the installation are the printing of the documentation, i.e. KSD, and running example programs from the command line in a ‘DOS’ window in Windows 95 or in a command window in Windows NT or under Unix. As indicated above, one may need to install GiveWin to see the graphs online.

4. DOCUMENTATION AND SAMPLE PROGRAMS

Documentation is often a weak part of modern computer packages, but SsfPack 2.2 is well documented and tested. It sets a standard which is far better than is available for many commercial products, not only scientifically. However, it uses a quite extensive statistical vocabulary, which may deter users who only have a background in regression and linear time series analysis. Many of the statistical terms are explained in Cox and Hinkley (1974), more recent developments are discussed in Barndorff-Nielsen and Cox (1994), but a deep knowledge is not always required to understand the main arguments.

The documentation is available in postscript format (.ps), which is printable with a program like Ghostview, and online searchable versions in portable document format (.pdf) will become available. Sections 1 to 6.3 of KSD are definitely worth reading and studying. Furthermore, I found the table in the Appendix extremely useful. One should note that KSD does not provide all the program listings in full.

In order to understand the example programs, which is essential for a productive use of SsfPack, one may need to consult the Ox documentation. The Ox documentation is available in the ‘browseable’ hypertext markup language (.html). As indicated above, OxEdit is a very useful program to search and browse the available example programs.

SsfPack was originally designed for estimation and simulation of so-called structural time series models, which are immediately written in state space form, see Section 3.2. The extra procedures to put ARMA and regression models in state space form, see Sections 3.1, 3.3, 5.1 and 6.3, now allow one to compare these widely used ‘reduced form’ models on the same footing, using the same recursive estimation structure, using the same optimal treatment of missing observations and using the same ‘exact’ estimation and simulation procedures.

© Royal Economic Society 1999
Exact results are not expected in all cases: as indicated in Sections 2.3 and 7, the ‘diffuse’ initial conditions of SsfPack do not always lead to exact likelihoods given the ad hoc solution presented in Section 2.3. An interesting application of the diffuse initial conditions is found in the state space representation of the regression model in Section 3.3. In a Bayesian interpretation (cf. Section 6.5) the SsfPack algorithm always imposes some prior information on the state variables in the ‘diffuse’ initial conditions, although this will hopefully only affect the fifth or sixth significant digits in the results.

SsfPack is not developed as a set of procedures for regression or an ARMA package. If one does not need recursive estimation of the ARMA parameters, existing freely available procedures in other libraries are computationally far more efficient, see also Section 7 of KSD. The ARMA procedures are interesting for educational purposes and they are of course useful in combining ARMA models with specific state space models as shown in Sections 5 and 6.

4.1. Practical issues and learning by doing

Learning to use a new set of procedures often starts by running a set of informative sample programs. The sample programs in KSD serve this purpose well. Moreover, the examples provide extra motivation for ‘nonexperts’ to study more difficult models such as the parametric time-varying state space formulation of the spline function in Sections 3.4 and 6.2.

The sample programs in Section 5 provide important guidelines for inexperienced readers who are used to menu-driven modelling and little econometric programming. The programs show how to concentrate out a scale factor, how to compute certain starting values, how to reparametrize not freely varying parameters, how to compute standard errors from the observed information, how to fine-tune convergence criteria, how to combine recursive estimation of the likelihood and the score. This also shows that the scope for minor and fatal mistakes in practical applications is quite large!

The sample programs are as simple as possible and they work. They are not meant as example programs for general classes of models. Extension to higher order models usually involves changing several lines of code, but with basic econometric and programming knowledge adaptation and debugging takes only minutes. The programs are well written. The output is assumed to be self-explanatory, but this really requires the graphical view. The sample programs only involve examples with a single measurement equation. I hope the authors will develop simple multivariate examples in future versions.

Easy simulation is essential for a good understanding of statistical models and the corresponding parametric inference. This is nicely shown by the simple examples in Section 4.2 for the sample paths of the local linear trend model and by the advanced example in Section 6.5 with the posterior inverse gamma distributions for the variances of state innovations and the measurement equation errors. Simulation-based estimation of nonGaussian state space models seems still too time consuming for learning by doing. Hopefully these techniques will become easier to evaluate in the coming years using advanced SsfPack versions.

The empirical examples also show other important steps in the time series modelling process: seasonal modelling, detection and treatment of outliers (as missing observations), and testing for parameter stability.

All in all, the programs provide a good basis for learning by doing, both at a basic and an advanced level, and I think that SsfPack is not only useful after acquiring the proper econometric skills, it is also useful in acquiring these skills.
5. EXTENSIONS

The authors and co-authors of cited papers will certainly generalize SsfPack to more ready-to-use non-Gaussian models, which often involves simulation-based estimation, and to multivariate models such as structural model based co-integration analysis so as to make these techniques available for a wider range of readers.

One can also extend SsfPack to simplify its use for particular applications. I have some experience in developing a successful application for ‘online’ day-to-day forecasting.

Since Ox is an object-oriented language it is not difficult to derive your own classes using SsfPack and the existing classes in Ox. Combining SsfPack and Ox’s Database class allows one, for example, to derive programs for data in Microsoft Excel files where variables and sample periods can be selected using variable names and calendar time. This does not require a lot of extra programming and unnecessary extra documentation.

One can imagine that one particular feature of SsfPack, or for that matter, Ox or GiveWin is too restrictive for the application one has in mind. In that case it is not very hard, although rather labour intensive, to link procedures of SsfPack via Ox in a Visual Basic or a Visual C++ program. In particular, this enables one to develop one’s own interface, e.g. for a menu-driven modelling and forecasting program for a certain class of state space models. Although it is an attractive option one should be prepared to invest weeks of labour to achieve significant results. The object-oriented nature of Ox and C++ allow for division of labour in the performance of the different tasks.

An intermediate realistic option is to develop only input menus in Visual C++ and use GiveWin via Ox for graphical output. Details are provided in Doornik (1998). The user of such a program then only requires a basic knowledge of the Windows system to operate it, so that she or he can concentrate on acquiring the knowledge needed to interpret the output generated using SsfPack procedures.

6. CONCLUSION

SsfPack is a well documented very up-to-date package which promises a high return on the considerable investment required to use it. The reading and studying of the documentation provided by Koopman, Shephard and Doornik is very worthwhile in its own right. I hope SsfPack will help in bridging the gap between interested statistical time series modellers and the advanced specialists, so as to improve their interaction.

REFERENCES


© Royal Economic Society 1999
