JDemetra+, an open framework for seasonal adjustment

Jean Palate*

National Bank of Belgium, Brussels, Belgium, jean.palate@nbb.be

Abstract

Seasonal adjustment (SA) is an important method for the analysis of short term statistics. The two leading algorithms in that domain are Tramo-Seats (TS) and X12-Arima/X13-Arima-Seats (X12). Besides SA itself, the time series modelling features included in those programs are useful for numerous tasks like outliers' detection, forecasting or estimation of missing values. Taking into account the broad scope of those methods, we have developed a new set of object-oriented tools that can easily be plugged into in-house developments or that can be extended for new needs. The modules, which form JDemetra+, are completely written in Java and are available in open source form. In addition to the new implementations of TS and of X12, JDemetra+ has also been enriched to deal with related time series problems like structural models, non-standard SA methods, temporal disaggregation, benchmarking and business cycle analysis. All those problems are handled by means of a common state space framework, which can be used for other purposes. We will describe the architecture of JDemetra+ and we will give an overview of its main statistical features. Some of them will be illustrated by means of practical examples. The results provided by the original TS and X12 methods and by the new implementation will also be compared, using a set of real time series.

Key words: Java, State space models, Tramo-Seats, X12-Arima

1. Introduction

Seasonal adjustment (SA) is an important step in the official statistics architecture. To achieve more harmonised practices, the SA Steering Group (a Eurostat-ECB group of experts) has encouraged the development of a flexible software solution for SA to be used within the European Statistical System. JDemetra+ is the last output of that initiative. The software, which is built on a set of object-oriented libraries developed at the National Bank of Belgium, includes new implementations of the two leading SA algorithms - Tramo-Seats (TS) and X12-Arima (X12), along with numerous tools for time series analysis. It is released as Open Source on the Joinup platform[1].

The need for statistical tools that can be integrated in an efficient way in various environments and that can be extended to face new requirements has been the driving force behind the developments. We shall give in the first parts of this paper a general overview of JDemetra+; we shall consider in more detail the SA framework of TS and X12 as well as the State space framework (SSF), which is hugely used in the current implementation.

In the second part of the study, we shall focus on more practical aspects of the new software. Two small examples of use of the framework will be presented. In a last point, we shall shortly compare a specific aspect of the estimation procedures used in the original programs and in JDemetra+.

2. General overview of JDemetra+

JDemetra+ is essentially composed of two main blocks: the graphical interface and the underlying libraries, which can be used independently.

The graphical interface - the only part that ordinary users interact with - is based on the NetBeans platform. By default, it gives a unified access to TS and to X12. Simplified interfaces to a few other SA methods (basic structural models...) are also provided, mainly for exploratory purposes. Third party IT teams can develop new modules that can be plugged into the original software in a transparent way.

The other libraries are pure Java modules. They can be split into algorithmic packages - the core of the software, called "JTstoolkit" - and peripheral services, like the access to external data providers or the formatting of the outputs. JTstoolkit provides low-level object-oriented services relating to common concepts of the time series domain, and more especially relating to the concepts handled in TS and in X12. Consequently, JDemetra+

---

[1] http://joinup.ec.europa.eu/software/sat

contains a "toolkit" for time series analysis, on which the new implementations of TS and of X12 are built and on which researchers/developers may draw to solve other problems. We will focus in this paper on that part of the software, which can be used in a completely independent way.

3. The seasonal adjustment framework

TS and X12 are both composed of two main steps. The first one consists in the modelling of the series by means of a regression model with ARIMA noises (hereafter REGARIMA). The regression model may include outliers, calendar variables or any user-defined variables. The original series may also be transformed before modelling. The second part of the algorithms will decompose the "linearized"[2] series into its main components (trend-cycle, seasonal and irregular components), either through a model-based approach (SEATS) or by means of predefined filters (X11).

In their pre-processing step, TS and X12 act as expert systems for discovering a suitable REGARIMA model. The method, initially developed in TS, mainly consists in an iterative processing which involves the following functional blocks:
- Initial analysis of the series, for detecting the presence or not of seasonality.
- Transformation of the series (logs or levels).
- Test on the presence of calendar effects and of other pre-specified regression variables.
- Automatic ARIMA modelling decomposed in the identification of the differencing orders and in the identification of the stationary ARMA orders.
- Automatic detection of outliers
- Final estimation of the model
- Control of the final model
- Comparison of the chosen model with a reference model

Although it follows a similar design, X12 will use in the different blocks slightly different decision tests. Depending on the options, the versions of the software or the expert system itself (TS or X12), some blocks may be skipped. The Java implementation in JDemetra+ provides modules (classes) that correspond to the different blocks of the two original programs. Those blocks can be interchanged, replaced or used elsewhere for their specific interest.

Considering the decomposition step, JDemetra+ offers for the heuristic X11 method a code based on the description of Ladiray and Quenneville (2001).
The libraries also provide several tools for the estimation of the model-based decomposition (UCARIMA models, estimated in SEATS)
- The Burman's algorithm (1980)
- A Kalman smoother, following Durbin and Koopman (2001, hereafter DK)
- Direct matrix computation, like in McElroy(2008)

They also contain analysis tools derived from the Wiener-Kolmogorov approach, as developed in Gomez and Maravall (2001) and implemented in SEATS.

All those algorithms can be re-used in a straightforward way in other decomposition schemes.

4. Common state space framework

In JDemetra+, state space models and their associated Kalman filters (KF) are key elements for the estimation of the REGARIMA models. They also provide efficient and stable solutions not only in the usual (SA) decomposition models but also in many other

---

[2] Series corrected by the effects of the regression variables.

problems, like benchmarking[3] or alternative SA methods. See for instance Bell (2004) for a few practical examples.

The state space framework (SSF) developed in the new tool tries to make a synthesis of the KF algorithms used in Tramo and of the more recent theoretical developments of DK. Some features related to the array algorithms described in Kailath et al (1999) have also been integrated. The SSF should play a central role in future extensions of the software.

We shortly present below the model supported by the libraries, its related algorithms and its current implementations.

The general linear Gaussian state-space model can be written in many different ways. We use a simplified version of the presentation of DK, in which the measurement and state equations are the following:

$$y_t = Z_t \alpha_t$$
$$\alpha_{t+1} = T_t \alpha_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, \sigma^2 V_t) \tag{1}$$

y_t is the observation at period t, α_t is the state vector. It should be noted that the measurement equation does not contain residuals. As the residuals can always be added to the state vector, the simplification is not a restriction. It leads to a cleaner code, at the expense of a slight performance penalty.

The residuals ε_t of the state equation will be modelled as:

$$\varepsilon_t = R_t \mu_t \qquad\qquad \mu_t \sim N(0, \sigma^2 Q_t) \tag{2}$$

where Q_t is a non-singular matrix. In other words, $V_t = R_t Q_t R_t'$.

The initial conditions of the filter are defined as follows:

$$\alpha_0 = a_0 + B\delta + \nu \qquad \begin{matrix} \nu \sim N(0, \sigma^2 P_*) \\ \delta \sim N(0, \kappa I) \end{matrix} \tag{3}$$

where κ is arbitrary large. P_* is the variance of the stationary part of the initial state vector and $P_\infty = BB'$ models the diffuse part.

Besides some basic atomic models (ARIMA, UCARIMA, Structural models...), the framework provides facilities for combining them (composite models, temporal aggregation models, extension by regression variables...) and numerous algorithms for filtering and smoothing (ordinary filtering and smoothing, fast Chandrasekhar recursions, disturbance smoothing, diffuse initialization following DK or the augmented Kalman filter...).

For high performances, the framework avoids as much as possible direct matrix computations by using "functional forms"[4]. Such a solution implies a huge use of object-oriented facilities, like abstract interfaces or inheritance. In that way, external developers will be able to enrich the framework with new models that will take advantage of all the current features (algorithms, performances, combination with existing models...).

5. Applications
One of the goals of the project is the development of reusable and extensible modules. We present below - without any detail - two straightforward applications of this principle, available in the graphical interface.

5.1 Wiener-Kolmogorov filters
The Wiener-Kolmogorov approach developed in SEATS can be applied in a transparent way to any UCARIMA models. The charts below give the pseudo-spectrums of the final

---

[3] For SA, benchmarking is used in the reconciliation of seasonally adjusted series with their raw annual totals.

[4] For instance, each model will be asked to transform a state vector in a way that corresponds to the transition equation.

(or historical) estimators of the different components (see Gomez and Maravall), for an Airline model and for a Generalized Airline model (see Aston, Findley et al (2004) for further details). The pseudo-spectrum of the seasonal component clearly shows the presence of a 6-months periodic behaviour in the Belgian imports of footwear. The generalized airline model will give, in some very specific cases, an interesting alternative to the traditional models.

Belgium. Imports of footwear (monthly series, 1995-2007)
Pseudo-spectrum of the final estimators (Wiener-Kolmogorov filters)

Airline model   Generalized airline model
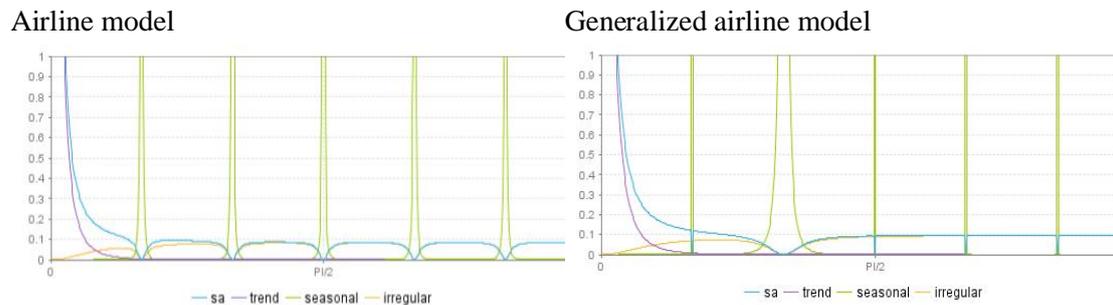
5.2 Kalman smoother

In TS, the calendar effects and the stochastic seasonal component are estimated separately: the former is measured in the pre-processing step and the latter in the decomposition step. Consequently, we do not have an exact estimation of the standard deviations of the final seasonal component. By extending the UCARIMA model with the regression variables, we can, by using the Kalman smoother, easily obtain such results. The chart below compares the standard deviations of the seasonal component as estimated in SEATS (separate) and as they appear in the full model (stochastic and final=stochastic+calendar).

Belgium. Imports of goods (monthly series, 1995-2007)
Standard deviations of the seasonal components (in logs; includes forecasts)
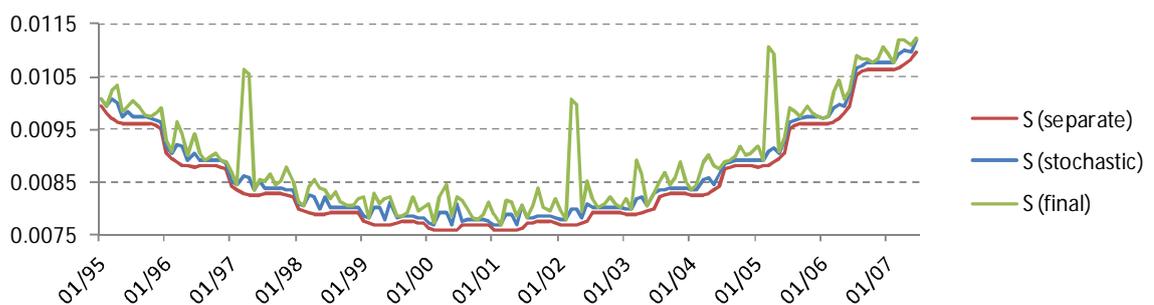
6. Comparison of Tramo-Seats, X12-Arima and JDemetra+

The development of a new implementation of TS and of X12 has raised a lot of discussion on the comparability of the results. The fact that JDemetra+ rests on a complete re-factoring of the original programs, sometimes with other algorithmic choices, has reinforced the (justified) caution of the users.

From a logical point of view, TS and X12 handle two problems of a different nature. On the one hand, they provide well-defined methods for decomposing a series of which the model has been completely identified. On the second hand, they are expert systems which are able to identify a suitable model for a series, following rather implicit criterions. We would expect from a new implementation that it could at least produce for the fully identified decomposition problem the same results as the original software (up to acceptable numerical differences). Considering the expert system features, JDemetra+

should be able - to be called "new implementation of TS and of X12" - to identify the same model in a large majority of the cases. Taking into account the lack of explicit criterions to compare models and the thresholds effects that are inherent to that kind of problem, a perfect comparability is rather unlikely.

We will focus in this short study on the actual estimation of the parameters of a well-defined REGARIMA model, which is a problem on the frontier between the expert system and the SA decomposition itself.

A systematic comparison of the results provided by X12 and TS revealed that, for complex models like the (3,1,1)(0,1,1), which is the last resort model in their automatic identification procedure, the original programs themselves lead to different results in more than 15% of the cases.

An in-depth analysis of the problem showed - not surprisingly - that the problem mainly appears in models that contain both auto-regressive and moving average parameters. Their likelihood function often contains several local maxima that the different optimization procedures will choose in a rather unpredictable way. In such cases, the initial estimate seems to be the more decisive element for the final result.

The optimization procedure used in JDemetra+ has been adapted to take into account those considerations. Grossly speaking, for models that mix auto-regressive and moving average parameters, the software tries different starting points that mainly differ by the dominant root in their regular moving average polynomial. After a few optimization steps with the different starting points, the "best" model is selected and the optimization procedure is completed.

Below we present the results obtained for a set of 300 monthly series on the German industrial production (from 1991 to 2009, source Eurostat), which is well representative.

Table 1

Comparison between:	JD+ / TS			JD+ / X12			TS / X12		
Model: Arima [+calendar effects]	JD+ better[5]	TS better	=	JD+ better	X12 better	=	TS better	X12 better	=
(0,1,1)(0,1,1)+TD7	0%	0%	100%	0%	0%	100%	0%	0%	100%
(1,1,1)(1,1,1)+TD7	**3%**	0%	97%	**2%**	0%	98%	0%	**2%**	98%
(2,1,1)(0,1,1)+TD7	**4%**	1%	95%	**7%**	0%	93%	**4%**	1%	95%
(3,1,1)(0,1,1)	**18%**	2%	80%	**11%**	1%	88%	6%	**11%**	83%
(1,1,3)(0,1,1)	**19%**	1%	80%	**14%**	1%	85%	5%	**11%**	83%

It should be stressed that the initial version of JDemetra+ did not perform better than the other algorithms. The significant improvement comes from the modification which is shortly described above.

The different local maxima that can be found for a given model may correspond to significantly different parameters. They will quite often present rather similar properties, so that their impact on the final SA results will be limited. However, that optimistic observation should not be considered as a rule.

The automatic model identification procedure of Tramo and of X12 will select complex models in a rather limited number of cases (> 3 parameters for 24% of the above series with Tramo, for 7% of the series with X12). Consequently, the above comparison exaggerates the problem. Anyway, in their iterative search for a suitable solution, the expert systems will consider at several points potentially unstable models. Although

---

[5] "Better" means significantly higher likelihood (and thus different estimates).

further investigation is needed, it seems that the estimation problem would explain a significant number of divergent solutions[6].

Finally, it should be mentioned that, in terms of seasonal adjustment, the consequences may be much more important for Tramo-Seats than for X12-Arima.

7. Conclusion

More than a simple translation, JDemetra+ is a re-factoring of TS and of X12 in an open object-oriented framework. The concepts and the algorithms of the original programs have been reorganized and extended in a rich toolbox on which the new implementation of the two leading SA algorithms is built. The libraries have also been designed for the developments of efficient solutions to other related time series problems.

Obtaining coherent results with the current official programs is certainly one of the main concerns of JDemetra+. First tests have revealed that the topic is complex and that it needs further investigation.

Last but not least, by using a modular architecture in a standard technology and by choosing an Open Source solution, we have set up the conditions for the participation of external teams to the project.

References

Aston, J. A. D., Findley, D. F., Wills, K. C., and Martin, D. E. K. (2004), "Generalizations of the Box-Jenkins Airline Model With Frequency-Specific Seasonal Coefficients", Proceeding of the NBER/NSF Time Series Conference.

Bell, W.R. (2004), "On RegComponent Time Series Models and Their Applications", in State Space and Unobserved Component Models, A.Harvey, S.J.Koopman, and N.Shephard (eds). Cambridge: Cambridge University Press.

Burman, J.P. (1980), "Seasonal adjustment by signal extraction", Journal of the Royal Statistical Society, Serie A. 143, 321-337.

De Jong P. and Chu-Chun-Lin S. (1994), "Stationary and Non-Stationary State Space Models", Journal of Time Series Analysis, 15, 2, 151-166.

Durbin J, and Koopman S.J. (2001), "Time Series Analysis by State Space Methods", Oxford University Press, Oxford.

Findley D.F. (2005), "Some Recent Developments and Directions in Seasonal Adjustment", Journal of Official Statistics, 21, 2, pp. 343–365.

Gomez V. and Maravall A. (1994), "Estimation, Prediction and Interpolation for Nonstationary Series with the Kalman Filter", Journal of the American Statistical Association, 89, 611-624.

Gomez V. and Maravall A. (2001), "Seasonal Adjustment and Signal Extraction in Economic Time Series", Ch. 8 in D. Peña, G.C. Tiao and R.S. Tsay, eds., "A Course in Time Series Analysis", New York: J. Wiley and Sons, 202-246.

Harvey AC (1989), "Forecasting, Structural Time Series Models and the Kalman Filter", Cambridge University Press, Cambridge.

Kailath T., Sayed A.H. and Hassibi B. (1999), "State-Space Estimation", Prentice-Hall, Englewood Cliffs, NJ.

Ladiray D. and Quenneville B. (2001), "Seasonal Adjustment With the X-11 Method", Lecture Notes in Statistics, 158, Springer.

McElroy T. (2008), "Matrix formulas for nonstationary ARIMA signal Extraction", Econometric Theory, 24, pp 988-1009.

Palate Jean (2005), "Reusable components for benchmarking using Kalman filters", Working Papers and Studies, European Communities.

Proietti T. (2000), "Comparing seasonal components for structural time series models", International Journal of Forecasting, 16, 2, 247-260.

---

[6] To understand why estimation diverges, it is necessary to compare the different implementations step by step. Such an analysis can't be automated, so that the actual sources of the discrepancies can't be easily assessed.