State Space Modeling Using **SAS**

Rajesh Selukar

SAS Institute

Abstract

This article provides a brief introduction to the state space modeling capabilities in SAS, a well-known statistical software system. SAS provides state space modeling in a few different settings. SAS/ETS, the econometric and time series analysis module of the SAS system, contains many procedures that use state space models to analyze univariate and multivariate time series data. In addition, SAS/IML, an interactive matrix language in the SAS system, provides Kalman filtering and smoothing routines for stationary and nonstationary state space models. SAS/IML also provides support for linear algebra and nonlinear function optimization, which makes it a convenient environment for general-purpose state space modeling.

Keywords: Kalman filter, state space model, ARIMA, unobserved components, SAS/ETS, SAS/IML.

1. Introduction

SAS software (SAS Institute Inc. 2008a) offers a set of solutions for enterprise-wide business users and provides a powerful fourth-generation programming language for performing tasks such as data management, report writing and graphics, statistical and mathematical analysis, and operations research. This article provides a brief introduction to the state space modeling capabilities that are available in SAS/ETS (SAS Institute Inc. 2010), the econometric and time series analysis module of the SAS system, and in SAS/IML (SAS Institute Inc. 2008b), the SAS interactive matrix language. The article is organized as follows. Section 2 uses the SAS/ETS UCM procedure to analyze the Nile data, the common example used in all articles in this volume. Section 3 discusses additional SAS/ETS procedures that are useful for state space modeling. Section 4 illustrates the use of SAS/IML for state space modeling. The final section contains concluding remarks.

2. Unobserved components modeling

The SAS/ETS UCM procedure can be used to analyze and forecast univariate time series data using an unobserved components model (UCM). A UCM decomposes the response series into components such as trend, seasonals, and the regression effects due to predictors. The next section uses the Nile data to illustrate some of these aspects of unobserved components modeling.

2.1. Analysis of Nile data

The Nile data consist of yearly readings of the Nile river water level, from 1871 to 1970, recorded at Aswan, Egypt. The analysis of this series begins by first fitting to it the *local level* model:

$$y_t = \mu_t + \epsilon_t \qquad \mu_t = \mu_{t-1} + \xi_t$$

That is, the readings y_t are noisy observations of the underlying level μ_t, which follows a random walk (RW). The observation noise ϵ_t and the RW disturbance ξ_t are assumed to be independent, Gaussian, white noise sequences with variance parameters σ_ϵ^2 and σ_ξ^2, respectively. The local level model is an example of a UCM that decomposes the response series into two unobserved components: the *level* component μ_t and the *irregular* component ϵ_t. It is a natural starting point in this analysis because the yearly water level of a large river, in the absence of major external shocks, can be expected to remain relatively constant for a long time. You can fit this model to the Nile data by using the UCM procedure as follows:

```
proc ucm data = nile;
   id year interval = year;
   model waterlevel;
   irregular plot = smooth;
   level checkbreak plot = smooth;
   estimate plot = residual;
   forecast plot = forecasts lead = 10 alpha = 0.5;
run;
```

The `PROC UCM` statement signifies the start of the UCM procedure and specifies the input data set, `nile`, which contains the response series. The optional `ID` statement specifies a date, time, or datetime variable (`year` in this example) to label the observations. The `INTERVAL = YEAR` option in the `ID` statement indicates that the measurements are collected on a yearly basis. The model specification begins with the `MODEL` statement, where the response series is specified (`waterlevel` in this case). The components in the model are specified using separate statements that enable you to control their individual properties. The irregular component, ϵ_t, is specified using the `IRREGULAR` statement, and the level component, μ_t, is specified using the `LEVEL` statement. The options in the `ESTIMATE` statement control the model-fitting environment and the diagnostics output—for example, Figure 1 shows the residual plot produced by using the `PLOT = RESIDUAL` option. The parameters of this model are the variances σ_ϵ^2 and σ_ξ^2; Table 1, produced by default, shows their maximum likelihood estimates. The options in the individual component statements serve different purposes—some determine their stochastic properties while others can be used to control their output. For example, Figure 2 shows the plot of the smoothed estimate of level μ_t, which is obtained by using the `PLOT =`
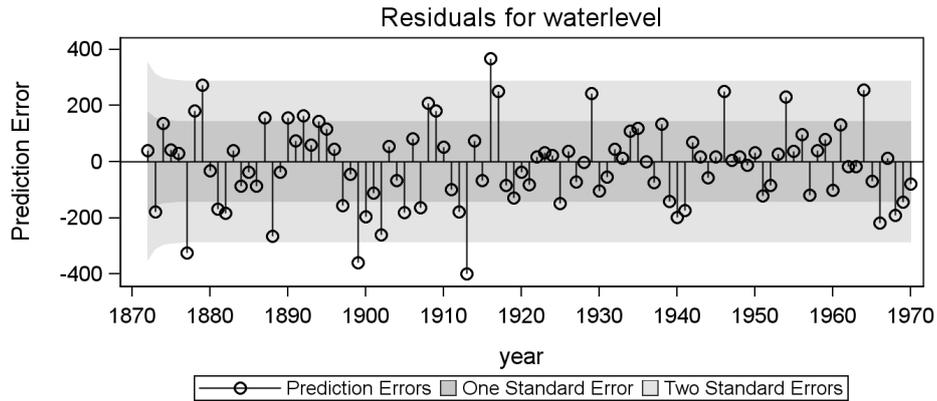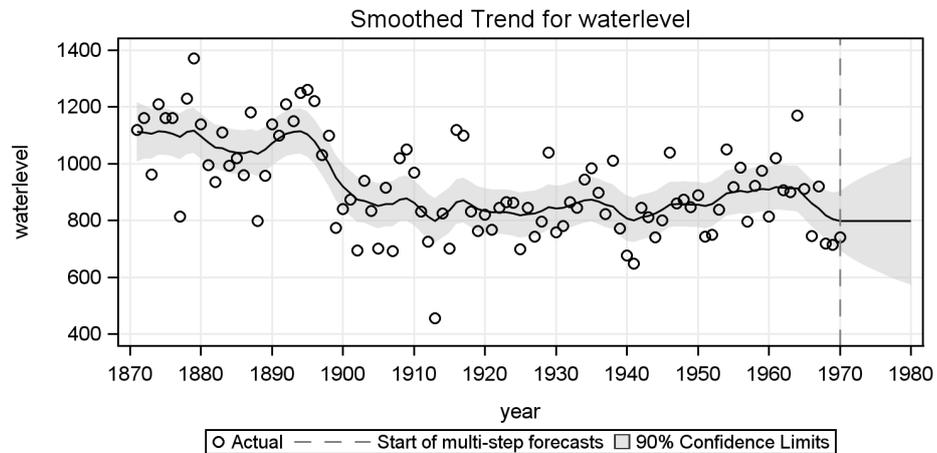
Figure 1: Local level model: One-step-ahead residuals.

| Component	Parameter	Estimate	Std Error
Irregular	Error Variance	15098.52	3145.55
Level	Error Variance	1469.17	1280.37

Table 1: Parameter estimates for the local level model.

Figure 2: Local level model: Smoothed μ_t.

SMOOTH option in the LEVEL statement. The plot seems to indicate a shift in the water level starting at or around 1899. This shift could be attributed to the start of construction of a dam near Aswan in that year. Here it is known that a shift in the water level occurred within the span of the series. However, in many cases such prior information is not available and it is useful to detect such shifts in a data analytic fashion. Table 2 shows a summary of the most likely outliers in the series based on the fitted model. Additive outliers are reported by default; in addition, level shifts are also reported if the CHECKBREAK option is used in the LEVEL statement. The outlier summary shows that the level shift in the year 1899 is statistically

Year	Break Type	Estimate	Std Error	Chi-Square	DF	Pr > ChiSq
1899	Level	-315.74	97.64	10.46	1	0.0012
1913	Additive Outlier	-406.02	133.60	9.24	1	0.0024

Table 2: Outlier summary.
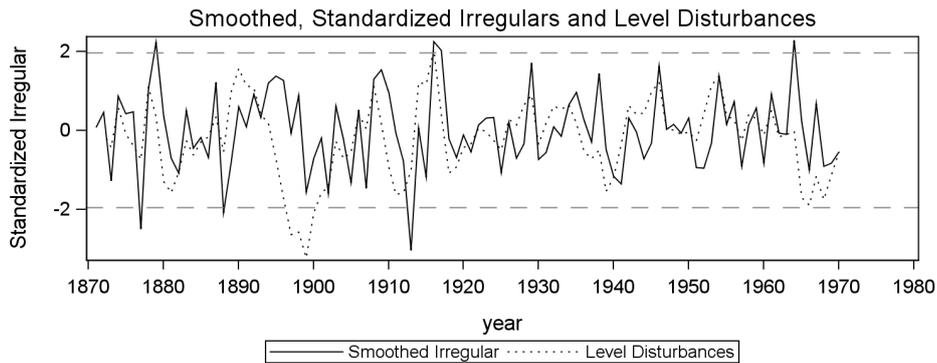
Figure 3: Local level model: Smoothed and standardized ϵ_t and ξ_t.

significant.

An additive outlier, an unusually low water level in 1913, is also detected. The outlier summary reported in Table 2 can also be inferred from the plot of smoothed, standardized auxiliary residuals shown in Figure 3. The series forecasts and the series decomposition can be requested using the FORECAST statement; the forecast plot shown in Figure 4 is produced by the PLOT = FORECASTS option.

The local level model can be corrected for the level shift of 1899 by including a regression variable, say ls1899, that is zero before 1899 and one thereafter. The additive outlier at 1913 can be similarly incorporated in the model by including a variable that is one at 1913 and zero elsewhere. In order to keep the model structurally simple the following program corrects the model for the level shift only. The level shift variable (ls1899) is specified as a regressor variable in the MODEL statement.

```
proc ucm data = nile;
    id year interval = year;
    model waterlevel = ls1899;
    irregular;
    level;
    estimate plot = panel;
    forecast plot = decomp lead = 10 alpha = 0.1;
run;
```

Table 3 shows the parameter estimates of this revised model. Note that the estimate of the noise variance σ_ϵ^2 has not changed much. However, the estimate of the level disturbance variance σ_ξ^2 has become nearly zero, indicating that the level component, after accounting
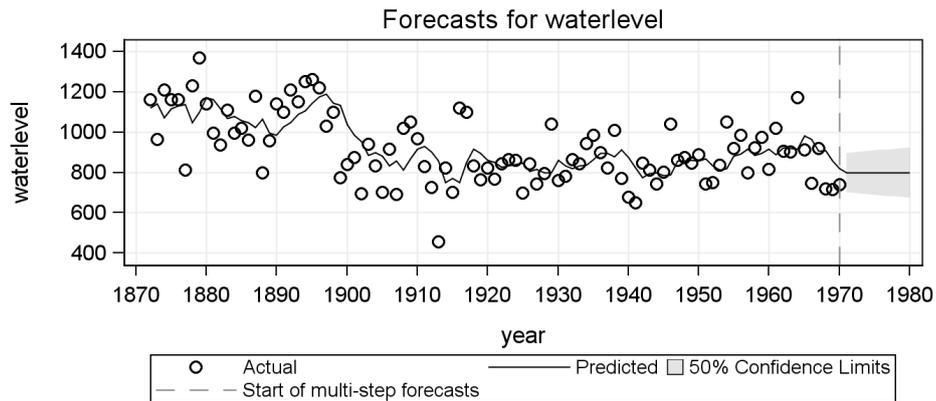
Figure 4: Local level model: Nile water level forecasts.

Component	Parameter	Estimate	Std Error
Irregular	Error Variance	16300.570	2328.66
Level	Error Variance	0.001	0.59
LS1899	Coefficient	−247.779	28.44

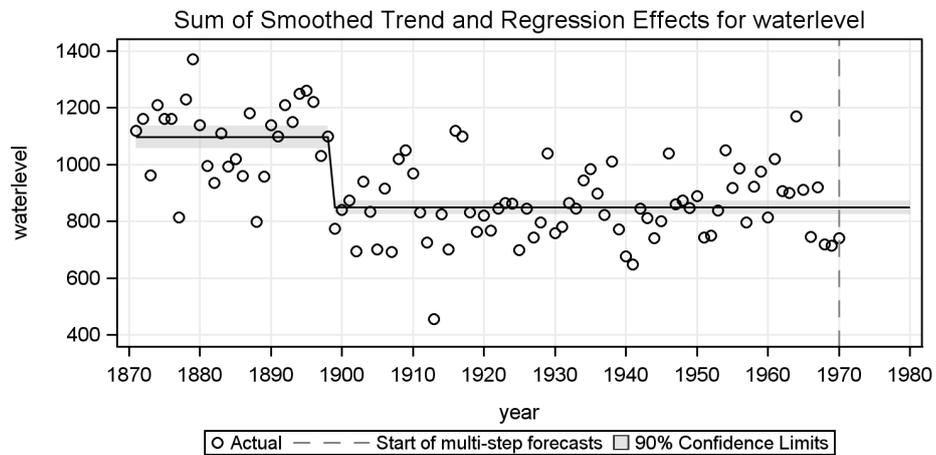Table 3: Parameter estimates for the local level plus LS1899 model.

Figure 5: Local level plus LS1899 model: Smoothed ($\mu_t + $ LS1899).

for the level shift of 1899, is nearly time-invariant. This is also evident from Figure 5, which shows the plot of the sum of smoothed trend and regression effect.

Finally, Figure 6, obtained by using the `PLOT = PANEL` option in the `ESTIMATE` statement, shows a panel of residual diagnostic plots that are useful for checking the normality and the lack of autocorrelation in the residuals. Figure 6 does not show any major violations of the model assumptions.
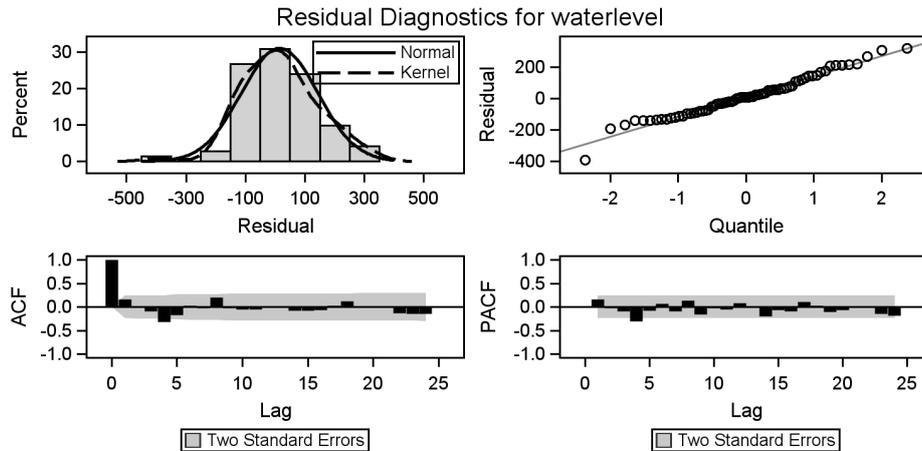
Figure 6: Local level plus LS1899 model: Residual diagnostics.

In conclusion, you can say that the local level model with the shift of 1899 is a reasonable description for the Nile data.

2.2. UCM procedure: Additional information

The preceding section provided a simple illustration of UC modeling using the UCM procedure. You can use the UCM procedure to fit a wide range of UCMs that can incorporate complex trend, seasonal, and cyclical patterns and can include multiple predictors. For example, you can analyze most univariate models discussed in Harvey (1989)—a classic reference for UC modeling. Moreover, more elaborate models such as the one proposed for modeling hourly electricity demand in Harvey and Koopman (1993) and a model proposed for seasonal adjustment of weekly data in Harvey, Koopman, and Riani (1997) can also be easily handled. Specifically, you can specify UCMs that consist of the following features:

- a variety of trend models, including random walk and local linear trend,

- several types of seasonals, including trigonometric seasonals with full control over the included harmonics and spline approximation for long seasonal patterns,

- multiple stochastic cycles,

- regression effects with time-invariant and time-varying regression coefficients,

- nonlinear regression effects that are accomplished using spline specification,

- an irregular component that can incorporate serial correlation modeled by a two-factor stationary ARMA process $(\mathrm{ARMA}(p,q)(P,Q)_s)$,

- models that include lagged response values – for example, an $\mathrm{ARIMA}(p,d,q)(P,D,Q)_s$ model.

The UCM procedure properly handles missing values in the response series. In particular, it provides a rigorous treatment of ARIMA models with missing data (see Kohn and Ansley 1986).

The UCMs considered in the UCM procedure are special cases of the (linear) Gaussian state space models (GSSM). The analysis of these GSSMs is based on the Kalman filtering and smoothing (KFS) algorithm. If the initial state of a GSSM is diffuse, as is often the case for UCMs, its treatment requires modification of the traditional KFS. The modification is called the diffuse KFS (DKFS). The DKFS implementation in the UCM procedure is based on de Jong (1991). The outlier detection process implemented in the UCM procedure is based on de Jong and Penzer (1998). For additional information, see the chapter "The UCM Procedure" in SAS Institute Inc. (2010).

3. State space modeling in **SAS/ETS**

Many state space models can be formulated as univariate or multivariate autoregressive moving average (ARIMA) models; for examples see Brockwell and Davis (1991) and Reinsel (1997). The SAS/ETS ARIMA and VARMAX procedures provide state-of-the-art facilities for univariate and multivariate ARIMA modeling, respectively.

The SAS/ETS STATESPACE procedure can be used for state space modeling of multivariate time series, including automatic model identification based on the canonical correlation analysis. The canonical correlation analysis implemented in the STATESPACE procedure is based on Aoki (1990). This procedure, however, is not a general purpose state space modeling procedure. In particular, it cannot be used to fit multivariate structural models discussed in Harvey (1989).

4. State space modeling using **SAS/IML**

SAS/IML is an interactive matrix language that, among other things, provides support for linear algebra and nonlinear function optimization. It also provides Kalman filtering and smoothing routines for stationary and non-stationary state space models.

Consider the following state space model with a partially diffuse initial state:

$$\mathbf{y}_t = \mathbf{X}_t \boldsymbol{\beta} + \mathbf{H}_t \mathbf{z}_t + \boldsymbol{\epsilon}_t, \quad t \geq 1,$$

$$\mathbf{z}_{t+1} = \mathbf{W}_t \boldsymbol{\beta} + \mathbf{F}_t \mathbf{z}_t + \boldsymbol{\eta}_t, \quad t \geq 0,$$

$$\mathbf{z}_0 = \mathbf{a} + \mathbf{A}\boldsymbol{\delta}, \quad \text{and} \quad \boldsymbol{\beta} = \mathbf{b} + \mathbf{B}\boldsymbol{\delta}$$

where $\mathbf{z}_t$ is the state vector, $\mathbf{y}_t$ is the observation vector, and for $t \geq 0$,

$$\left[\begin{array}{c} \boldsymbol{\eta}_t \\ \boldsymbol{\epsilon}_t \end{array} \right] \sim N \left(\mathbf{0}, \sigma^2 \left[\begin{array}{cc} \mathbf{V}_t & \mathbf{G}_t \\ \mathbf{G}_t^\top & \mathbf{R}_t \end{array} \right] \right), \quad \boldsymbol{\delta} \sim N(\mu, \sigma^2 \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} \to \infty$$

The noise vectors $(\boldsymbol{\eta}_t^\top, \boldsymbol{\epsilon}_t^\top)^\top$ and the (diffuse) random vector $\boldsymbol{\delta}$ are assumed to be mutually independent. The system matrices, $\mathbf{W}_t$, $\mathbf{F}_t$, $\mathbf{X}_t$, $\mathbf{H}_t$, $\mathbf{a}$, $\mathbf{A}$, $\mathbf{b}$, $\mathbf{B}$, $\mathbf{V}_t$, $\mathbf{G}_t$, and $\mathbf{R}_t$, are assumed to be known. The analysis of state space models of this form requires modification of the traditional Kalman filtering and smoothing (KFS) algorithm, which is called the diffuse KFS. You can analyze state space models of this form by using the SAS/IML subroutine

Introductory paper	SAS/IML state space model
α_t	$\mathbf{z}_t$
T_t	$\mathbf{F}_t$
$R_t \eta_t$	$\boldsymbol{\eta}_t$
$R_t Q_t R_t^\top$	$\sigma^2 \mathbf{V}_t$
ε_t	$\boldsymbol{\epsilon}_t$
Z_t	$\mathbf{H}_t$
H_t	$\sigma^2 \mathbf{R}_t$

Table 4: Notation for state space models.

KALDFF for diffuse Kalman filtering, and by using the subroutine KALDFS for diffuse Kalman smoothing. The SAS/IML DKFS implementation is based on de Jong (1991). Note that the model described above can be recast into the canonical state space model described in the introductory paper Commandeur, Koopman, and Ooms (2011) of the current volume. This is done by incorporating the regression vector $\boldsymbol{\beta}$ into the state vector and, if state and observation noise sequences are correlated, by incorporating the observation noise into the state as well. Table 4 provides additional clarification of the difference in notation between the two representations.

The SAS/IML documentation contains detailed information on KALDFF and KALDFS subroutines, including examples of their use. These subroutines are organized such that fairly complex state space models can be easily specified and useful quantities needed in their analysis are obtained as output. A brief description of these routines is given below.

KALDFF subroutine has the following signature:

```
CALL KALDFF( pred, vpred, initial, s2, data, lead, int, coef,
var, intd, coefd, n0, at, mt, qt );
```

where the arguments have the following meaning:

- Input:

    - data contains the y-values.
    - lead is the number of steps to forecast after the end of the data.
    - int contains information about the regressors $\mathbf{X}_t$ and $\mathbf{W}_t$ ($t \geq 1$).
    - coef contains information about the system matrices $\mathbf{F}_t$ and $\mathbf{H}_t$ ($t \geq 1$).
    - var contains information about the system matrices $\mathbf{R}_t$, $\mathbf{V}_t$ and $\mathbf{G}_t$ ($t \geq 1$).
    - intd contains information about the system vectors $\mathbf{a}$ and $\mathbf{b}$.
    - coefd contains information about the system matrices $\mathbf{A}$ and $\mathbf{B}$.

- Output:

    - pred contains the one-step-ahead state forecasts.
    - vpred contains the covariances of the one-step-ahead state forecasts.

- **initial** contains the estimate of the initial state δ along with its covariance estimate.
- **s2** contains estimate of the scalar σ^2.

- Input/Output (these arguments act as input as well as output and are better explained in the documentation):

  - **n0** is an optional integer argument that contains divisor used for computing the estimate of σ^2.
  - **qt** is an optional argument that on output contains quantities needed for the likelihood computation.
  - **at** is an optional argument that on output contains certain intermediate quantities needed during the subsequent smoothing computations (**KALDFS** call). As input it is used to provide certain initialization information.
  - **mt** is an optional argument that on output contains certain intermediate quantities needed during the subsequent smoothing computations (**KALDFS** call) and for the likelihood computation. As input it is used to provide certain initialization information (such as $\mathbf{V}_0$).

The **KALDFS** subroutine computes the smoothed state vector and its mean square error matrix from the one-step-ahead forecast and mean square error matrix computed by **KALDFF**. It has the following signature:

```
CALL KALDFS( sm, vsm, data, int, coef, var, bvec, bmat, initial,
at, mt, s2 , un, vun ) ;
```

where the arguments have the following meaning:

- Input:

  - **data** contains the y-values.
  - **int** contains information about the regressors $\mathbf{X}_t$ and $\mathbf{W}_t$.
  - **coef** contains information about the system matrices $\mathbf{F}_t$ and $\mathbf{H}_t$.
  - **var** contains information about the system matrices $\mathbf{R}_t$, $\mathbf{V}_t$ and $\mathbf{G}_t$.
  - **bvec** contains information about the system vector $\mathbf{b}$.
  - **bmat** contains information about the system matrix $\mathbf{B}$.
  - quantities computed by the preceding **KALDFF** call:
    * **initial** contains the estimate of the initial state δ along with its covariance estimate.
    * **s2** contains estimate of the scalar σ^2.
    * **at**.
    * **mt**.

- Output:

  - **sm** contains the smoothed state vectors.

        – `vsm` contains the covariances of the smoothed state vectors.

- Input/Output:

        – `un` is an optional argument that on output contains certain quantities used in smoothing recursions.

        – `vun` is an optional argument that on output contains certain quantities used in smoothing recursions.

The attached program illustrates how to use SAS/IML to specify and fit a simple multivariate state space model. More general state space models, possibly with time-varying system matrices, can be handled in a similar fashion. The remainder of this section describes the model used in this illustration, a broad outline of the attached program, and the produced output.

Consider the following two-way random effects panel model: for $t \geq 1$,

$$\mathbf{y}_t = \mathbf{X}_t \boldsymbol{\beta} + \boldsymbol{\gamma} + \tau_t \mathbf{1} + \boldsymbol{\epsilon}_t$$

where $\mathbf{y}_t$ are k-dimensional observations, $\mathbf{X}_t$ are $k \times p$ dimensional matrices that contain predictor variables, $\boldsymbol{\beta}$ is a p-dimensional regression parameter vector, the cross-sectional effects vector $\boldsymbol{\gamma}$ is a k-dimensional zero-mean Gaussian random vector with covariance $\sigma_\gamma^2 I_k$, τ_t (which denote the time effects that are common across cross-sections) is a sequence of uncorrelated, zero-mean Gaussian random variables with variance σ_τ^2, and $\boldsymbol{\epsilon}_t$ is a sequence of uncorrelated, zero-mean k-dimensional random vectors with covariance $\sigma_\epsilon^2 \mathbf{I}_k$. Here $\mathbf{I}_k$ denotes a k-dimensional identity matrix, $\mathbf{1}$ denotes a vector with all entries equal to 1, and the random sequences $\boldsymbol{\epsilon}_t$ and τ_t, and the random vector $\boldsymbol{\gamma}$ are assumed to be mutually independent.

The preceding model can be written as a state space model as

$$\mathbf{y}_t = \mathbf{X}_t \boldsymbol{\beta} + \mathbf{H}\mathbf{z}_t + \boldsymbol{\epsilon}_t \quad \mathbf{z}_{t+1} = \mathbf{F}\mathbf{z}_t + \boldsymbol{\eta}_t$$

where the $(k+1)$-dimensional state vectors $\mathbf{z}_t$ consist of τ_t as the first element and $\boldsymbol{\gamma}$ as the remaining elements, the $k \times (k+1)$ matrix $\mathbf{H}$ is equal to $[\mathbf{1} \quad \mathbf{I}_k]$, the $(k+1) \times (k+1)$ matrix $\mathbf{F}$ is a diagonal matrix with all diagonal elements equal to one except the first (which is zero). For $t \geq 1$, the $(k+1)$-dimensional disturbance vectors $\boldsymbol{\eta}_t$ in the state equation consist of all zeros except the first element, which is τ_t. $\boldsymbol{\eta}_0$ is a zero mean Gaussian vector with diagonal covariance consisting of σ_τ^2 as its first diagonal element and σ_γ^2 as its remaining diagonal elements. The initial state, $\mathbf{z}_0$, is zero—that is, $\mathbf{a}$ and $\mathbf{A}$ are zero—and the diffuse vector $\boldsymbol{\delta}$ is the same as the regression parameter vector $\boldsymbol{\beta}$, which means $\mathbf{b}$ is zero and $\mathbf{B}$ is identity. The observation and state noise sequences are uncorrelated, that is, $\mathbf{G}_t$ are zero. The tuning parameters in this model are the disturbance variances $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2, which can be estimated by optimizing the likelihood of the data. The regression parameter vector $\boldsymbol{\beta}$, the cross-sectional effects $\boldsymbol{\gamma}$, and the time effects τ_t are estimated by using state smoothing.

The attached program uses the airline cost data set from Greene (2000) to illustrate how to use SAS/IML to specify and fit this state space model. In the illustration the panel model studies the dependence of the log transformed cost, `lC`, on three variables—the log-transformed quantity (`lQ`), the log-transformed price of fuel (`lPF`), and the load factor (`LF`). An overall intercept is also included in the model. The data set contains yearly data on six airlines, which form the panels, over the fifteen-year time span 1970–1984. Note that, for this example, the elements of $\boldsymbol{\gamma}$ signify airline specific corrections to the overall mean and a time effect τ_t represents a similar correction to the overall mean at time t.

The program can be broadly described by the following steps:

- A subroutine, `rantwo`, is defined that does filtering, smoothing, and likelihood computation for the two-way random effects panel model. This is done by creating appropriate system matrices corresponding to this model and calling the `KALDFF` and `KALDFS` subroutines to do the actual computations. More precisely, the signature of `rantwo` is as follows:

      CALL RANTWO(y, X, va, vb, ve, smooth, pred, vpred, initial,
      dff_logl, sm, vsm);

  where the arguments have the following meaning:

  - Input
      * `y` contains the y-values, for example, `lC` values for the six airlines in the case of this example
      * `X` contains the regressors, for example, the intercept column and the `lQ`, `lPF`, and `LF` values for the six airlines in the case of this example
      * `va`, `vb`, and `ve` contain the variances $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2
      * `smooth`, a zero or one flag for deciding whether the smoothing step is to be performed or not
  - Output
      * `pred` and `vpred` contain the one-step-ahead state predictions and their covariances
      * `sm` and `vsm` contain the smoothed states and their covariances
      * `initial` contains the estimate of $\boldsymbol{\beta}$ and its covariance
      * `dff_logl` contains the diffuse log-likelihood

- A very simple function, `rantwo_2ll`, is defined, which computes minus two times the log-likelihood of the data for given values of $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2, by calling `rantwo`. It assumes that the data, `y` and `X`, remain fixed (global variables).

- Obtain the maximum likelihood estimates of the variance parameters by minimizing `rantwo_2ll` with respect to $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2. This is accomplished by using a SAS/IML non-linear optimization subroutine `NLPQN`. SAS/IML contains a set of optimization subroutines for minimizing or maximizing a continuous nonlinear function. The parameters can be subject to boundary constraints and linear or nonlinear equality and inequality constraints.

- Obtain the estimates of the regression parameter vector $\boldsymbol{\beta}$, the cross-sectional effects $\boldsymbol{\gamma}$, and the time effects τ_t by state smoothing. This is done by calling `rantwo` at the optimized values of $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2.

Finally, a portion of the output from this program is shown next. This output has been slightly modified for readability purposes. Table 5 shows the maximum likelihood estimates of $\sigma_\gamma^2, \sigma_\tau^2$, and σ_ϵ^2. Similarly, Table 6 shows the estimates of regression effects and their standard errors, and Table 7 shows the smoothed estimate of the panel effect vector $\boldsymbol{\gamma}$.

σ^2_γ	σ^2_τ	σ^2_ϵ
0.0151623	0.0010679	0.002738

Table 5: ML estimates of the variance parameters.

Effect	Estimate	Std Error
Intercept	9.36913	0.234647
lQ	0.86744	0.024532
lPF	0.43595	0.016621
LF	-0.98529	0.218658

Table 6: Regression effects $\boldsymbol{\beta}$.

Airline 1	Airline 2	Airline 3	Airline 4	Airline 5	Airline 6
0.0666301	0.0116846	-0.202884	0.1521905	-0.041562	0.0139418

Table 7: Smoothed estimate of the panel effects vector $\boldsymbol{\gamma}$.

5. Conclusion

This article and the attached SAS program provide a brief overview of the state space modeling functionality available in SAS/IML and SAS/ETS. Generally speaking, if a state space model belongs to one of the standard forms such as ARIMA, VARMAX, or UCM, it is easier to use the SAS/ETS procedures that are specially designed for their handling. Otherwise, you can use SAS/IML software for general state space modeling. For additional information you can consult SAS online documentation available at http://support.sas.com/documentation/. There you can find SAS Institute Inc. (2010) for SAS/ETS documentation, and SAS Institute Inc. (2008b) for SAS/IML documentation.

References

Aoki M (1990). *State Space Modeling of Time Series*. 2nd edition. Springer-Verlag, New York.

Brockwell P, Davis R (1991). *Time Series: Theory and Methods*. 2nd edition. Springer-Verlag, New York.

Commandeur JJF, Koopman SJ, Ooms M (2011). "Statistical Software for State Space Methods." *Journal of Statistical Software*, **41**(1), 1–18. URL http://www.jstatsoft.org/v41/i01/.

de Jong P (1991). "The Diffuse Kalman Filter." *The Annals of Statistics*, **19**, 1073–1083.

de Jong P, Penzer J (1998). "Diagnosing Shocks in Time Series." *Journal of the American Statistical Association*, **93**, 796–806.

Greene WH (2000). *Econometric Analysis.* 4th edition. Macmillan Publishing Company, New York.

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

Harvey AC, Koopman SJ (1993). "Forecasting Hourly Electricity Demand Using Time-Varying Splines." *Journal of the American Statistical Association*, **88**, 1228–1236.

Harvey AC, Koopman SJ, Riani M (1997). "The Modeling and Seasonal Adjustment of Weekly Observations." *Journal of the American Statistical Association*, **15**, 354–368.

Kohn R, Ansley CF (1986). "Estimation, Prediction, and Interpolation for ARIMA Models with Missing Data." *Journal of the American Statistical Association*, **81**, 751–761.

Reinsel GC (1997). *Elements of Multivariate Time Series Analysis.* 2nd edition. Springer-Verlag, New York.

SAS Institute Inc (2008a). *The SAS System, Version 9.2.* SAS Institute Inc., Cary, NC. URL http://www.sas.com/.

SAS Institute Inc (2008b). *SAS/IML 9.2 User's Guide.* SAS Institute Inc., Cary, NC. URL http://support.sas.com/documentation/cdl/en/imlug/59656/PDF/default/imlug.pdf.

SAS Institute Inc (2010). *SAS/ETS 9.22 User's Guide.* SAS Institute Inc., Cary, NC. URL http://support.sas.com/documentation/cdl/en/etsug/60372/PDF/default/etsug.pdf.

Affiliation:

Rajesh Selukar
SAS Campus Drive
Cary, NC 27513, United States of America
E-mail: rajesh.selukar@sas.com