Bayesian Estimation for Parsimonious Threshold Autoregressive Models in R

by Cathy W.S. Chen, Edward M.H. Lin, F.C. Liu, and Richard Gerlach

Introduction

The threshold autoregressive (TAR) model proposed by Tong (1978, 1983) and Tong and Lim (1980) is a popular nonlinear time series model that has been widely applied in many areas including ecology, medical research, economics, finance and others (Brockwell, 2007). Some interesting statistical and physical properties of TAR include asymmetry, limit cycles, amplitude dependent frequencies and jump phenomena, all of which linear models are unable to capture. The standard two-regime threshold autoregressive (TAR) model is considered in this paper. Given the autoregressive (AR) orders p_1 and p_2, the two-regime TAR(2:p_1;p_2) model is specified as:

$$y_t = \begin{cases} \phi_0^{(1)} + \sum_{i=1}^{p_1} \phi_{k_i}^{(1)} y_{t-k_i} + a_t^{(1)} \text{ if } z_{t-d} \le r, \\ \phi_0^{(2)} + \sum_{i=1}^{p_2} \phi_{l_i}^{(2)} y_{t-l_i} + a_t^{(2)} \text{ if } z_{t-d} > r, \end{cases}$$

where $\{k_i, i = 1, \ldots, p_1\}$ and $\{l_i, i = 1, \ldots, p_2\}$ are subsets of $\{1, \ldots, p\}$,

where r is the threshold parameter driving the regime-switching behavior; where p is a reasonable maximum lag [1]; z_t is the threshold variable; d is the threshold lag of the model; $a_t^{(1)}$ and $a_t^{(2)}$ are two independent Gaussian white noise processes with mean zero and variance σ_j^2, $j = 1, 2$. It is common to choose the threshold variable $\mathbf{z}$ as a lagged value of the time series itself, that is $z_t = y_t$. In this case the resulting model is called Self-Exciting (SETAR). In general, $\mathbf{z}$ could be any exogenous or endogenous variable (Chen 1998). The TAR model consists of a piecewise linear AR model in each regime, defined by the threshold variable z_{t-d} and associated threshold value r. Note that the parameter p is not an input to the `R` program, but instead should be considered by the user as the largest possible lag the model could accommodate, e.g. in light of the sample size n. i.e. $p << n$ is usually enforced for AR models.

Frequentist parameter estimation of the TAR model is usually implemented in two stages; see for example Tong and Lim (1980), Tong (1990) and Tsay (1989, 1998). For fixed and subjectively chosen values of d (usually 1) and r (usually 0), all other model parameters are estimated first. Then, conditional on these parameter estimates, d and r can be estimated by: minimizing the AIC, minimizing a nonlinearity test statistic and/or using scatter plots (Tsay, 1989); or by minimizing a conditional least squares formula (Tsay, 1998).

Bayesian methods allow simultaneous inference on all model parameters, in this case allowing uncertainty about the threshold lag d and threshold parameter r to be properly incorporated into estimation and inference. Such uncertainty is not accounted for in the standard two-stage methods. However, in the nonlinear TAR setting, neither the marginal or joint posterior distributions for the parameters can be easily analytically obtained: these usually involve high dimensional integrations and/or non-tractable forms. However, the joint posterior distribution can be evaluated up to a constant, and thus numerical integration techniques can be used to estimate the marginal distributions required. The most successful of these, for TAR models, are Markov chain Monte Carlo (MCMC) methods, specifically those based on the Gibbs sampler. Chen and Lee (1995) proposed such a method, incorporating the Metropolis-Hastings (MH) algorithm (Metropolis *et al.*, 1953; Hastings, 1970), for inference in TAR models. Utilizing this MH within Gibbs algorithm, the marginal and posterior distributions can be estimated by iterative sampling. To the best of our knowledge, this is the first time a Bayesian approach for TAR models has been offered in a statistical package.

We propose an `R` package `BAYSTAR` that provides functionality for parameter estimation and inference for two-regime TAR models, as well as allowing the monitoring of MCMC convergence by returning all MCMC iterates as output. The main features of the package `BAYSTAR` are: applying the Bayesian inferential methods to simulated or real data sets; online monitoring of the acceptance rate and tuning parameters of the MH algorithm, to aid the convergence of the Markov chain; returning all MCMC iterates for user manipulation, clearly reporting the relevant MCMC summary statistics and constructing trace plots and auto-correlograms as diagnostic tools to assess convergence of MCMC chains. This allows us to statistically estimate all unknown model parameters simultaneously, including capturing uncertainty about threshold value and delay lag; not accounted for in standard methods that condition upon a particular threshold value and delay lag, see e.g. the SETAR function which is available in the "tsDyn" package at `CRAN`. We also allow the user to define a parsimonious separate AR order specification in each regime. Using our code it is possible to set some AR parameters in either or both regimes to be

[1] We have tried up to p=50 successfully.

zero. That is, we could set $p_1 = 3$ and subsequently estimate any three parameters of our convenience.

Prior settings

Bayesian inference requires us to specify a prior distribution for the unknown parameters. The parameters of the TAR(2:p_1;p_2) model are Θ_1, Θ_2, σ_1^2, σ_2^2, r and d, where Θ_1=($\phi_0^{(1)}$, $\phi_1^{(1)}$, ..., $\phi_{p_1}^{(1)}$)$'$ and Θ_2=($\phi_0^{(2)}$, $\phi_1^{(2)}$, ..., $\phi_{p_2}^{(2)}$)$'$. We take fairly standard choices: Θ_1, Θ_2 as independent $N(\Theta_{0i}, V_i^{-1})$, $i = 1, 2$, and employ conjugate priors for σ_1^2 and σ_2^2,

$$\sigma_i^2 \sim IG\left(\nu_i/2, \nu_i\lambda_i/2\right), \quad i = 1, 2,$$

where IG stands for the inverse Gamma distribution. In threshold modeling, it is important to set a minimum sample size in each regime to generate meaningful inference. The prior for the threshold parameter r, follows a uniform distribution on a range (l, u), where l and u are set as relevant percentiles of the observed threshold variable. This prior could be considered to correspond to an empirical Bayes approach, rather than a fully Bayesian one. Finally, the delay d has a discrete uniform prior over the integers: 1,2,..., d_0, where d_0 is a user-set maximum delay. We assume the hyper-parameters, $(\Theta_{0i}, V_i, \nu_i, \lambda_i, a, b, d_0)$ are known and can be specified by the user in our `R` code.

The MCMC sampling scheme successively generates iterates from the posterior distributions for groups of parameters, conditional on the sample data and the remaining parameters. Multiplying the likelihood and the priors, using Bayes' rule, leads to these conditional posteriors. For details, readers are referred to Chen and Lee (1995). Only the posterior distribution of r is not a standard distributional form, thus requiring us to use the MH method to achieve the desired sample for r. The standard Gaussian proposal random walk MH algorithm is used. To yield good convergence properties for this algorithm, the choice of step size, controlling the proposal variance, is important. A suitable value of the step size, with good convergence properties, can be achieved by tuning to achieve an acceptance rate between 25% to 50%, as suggested by Gelman, Roberts and Gilks (1996). This tuning occurs only in the burn-in period.

Exemplary applications

Simulated data

We now illustrate an example with simulated data. The data is generated from a two-regime SETAR(2 : 2; 2) model specified as:

$$y_t = \begin{cases} 0.1 - 0.4y_{t-1} + 0.3y_{t-2} + a_t^{(1)} \text{ if } y_{t-1} \le 0.4, \\ 0.2 + 0.3y_{t-1} + 0.3y_{t-2} + a_t^{(2)} \text{ if } y_{t-1} > 0.4, \end{cases}$$

$$\text{where } a_t^{(1)} \sim N(0, 0.8) \quad \text{and} \quad a_t^{(2)} \sim N(0, 0.5).$$

Users can import data from an external file, or use their own simulated data, and directly estimate model parameters via the proposed functions. To implement the MCMC sampling, the scheme was run for $N = 10,000$ iterations (the total MCMC sample) and the first $M = 2,000$ iterations (the burn-in sample) were discarded.

```
+ nIterations<- 10000
+ nBurnin<- 2000
```

The hyper-parameters are set as Θ_{0i} = **0**, V_i = diag(0.1, ..., 0.1), ν_i = 3 and $\lambda_i = \widetilde{\sigma}^2/3$ for i = 1,2, where $\widetilde{\sigma}^2$ is the residual mean squared error of fitting an AR(p_1) model to the data. The motivation to choose the hyper-parameters of ν_i and λ_i is that the expectation of σ_i^2 is equal to $\widetilde{\sigma}^2$. The maximum delay lag is set to d_0 = 3. We choose $a = Q_1$ and $b = Q_3$: the 1st and 3rd quartiles of the data respectively, for the prior on r.

```
+ mu0<- matrix(0, nrow=p1+1, ncol=1)
+ v0<- diag(0.1, p1+1)
+ ar.mse<- ar(yt,aic=FALSE, order.max=p1)
+ v<- 3; lambda<- ar.mse$var.pred/3
```

The MCMC sampling steps sequentially draw samples of parameters by using the functions `TAR.coeff()`, `TAR.sigma()`, `TAR.lagd()` and `TAR.thres()`, iteratively. `TAR.coeff()` returns the updated values of Θ_1 and Θ_2 from a multivariate normal distribution for each regime. σ_1^2 and σ_2^2 are sampled separately using the function `TAR.sigma()` from inverse gamma distributions. `TAR.lagd()` and `TAR.thres()` are used to sample d, from a multinomial distribution, and r, by using the MH algorithm, respectively. The required log-likelihood function is computed by the function `TAR.lik()`. When drawing r, we monitor the acceptance rate of the MH algorithm so as to maximize the chance of achieving a stationary and convergent sample. The `BAYSTAR` package provides output after every 1,000 MCMC iterations, for monitoring the estimation, and the acceptance rate, of r. If the acceptance rate falls outside 25% to 50%, the step size of the MH algorithm is automatically adjusted during burn-in iterations, without re-running the whole program. Enlarging the step size should reduce the acceptance rate while diminishing the step size should increase this rate.

A summary of the MCMC output can be obtained via the function `TAR.summary()`. `TAR.summary()` returns the posterior mean, median, standard deviation and the lower and upper bound of the 95%

```
                true    mean  median   s.d.   lower   upper
phi0^1      0.1000  0.0873  0.0880 0.0395  0.0096  0.1641
phi1^1     -0.4000 -0.3426 -0.3423 0.0589 -0.4566 -0.2294
phi2^1      0.3000  0.2865  0.2863 0.0389  0.2098  0.3639
phi0^2      0.2000  0.2223  0.2222 0.0533  0.1187  0.3285
phi1^2      0.3000  0.2831  0.2836 0.0407  0.2040  0.3622
phi2^2      0.3000  0.3244  0.3245 0.0234  0.2780  0.3701
sigma1      0.8000  0.7789  0.7773 0.0385  0.7079  0.8587
simga2      0.5000  0.5563  0.5555 0.0231  0.5132  0.6029
r           0.4000  0.4161  0.4097 0.0222  0.3968  0.4791
diff.phi0 -0.1000 -0.1350 -0.1354 0.0654 -0.2631 -0.0039
diff.phi1 -0.7000 -0.6257 -0.6258 0.0726 -0.7657 -0.4841
diff.phi2  0.0000 -0.0379 -0.0381 0.0455 -0.1256  0.0521
mean1       0.0909  0.0834  0.0829 0.0390  0.0088  0.1622
mean2       0.5000  0.5598  0.5669 0.0888  0.3673  0.7161
Lag choice :
         1 2 3
Freq 10000 0 0
```

Figure 1: The summary output for all parameters is printed as a table.

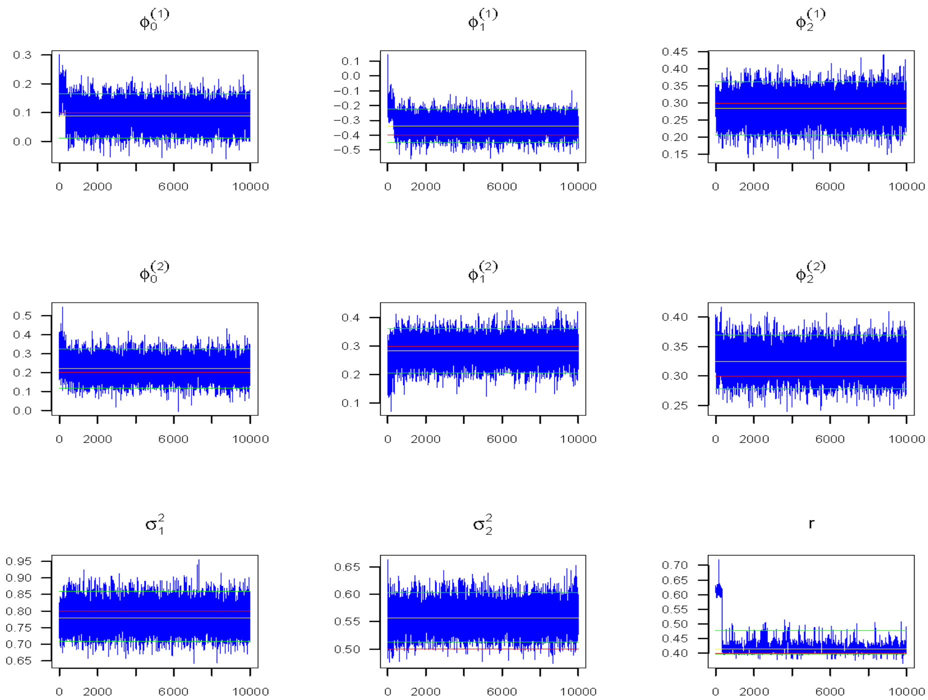

Figure 2: The trace plots of all MCMC iterations for all parameters.

Bayes posterior interval for all parameters, all obtained from the sampling period only, after burn-in. Output is also displayed for the differences in the mean coefficients and the unconditional mean in each regime. The summary statistics are printed as in Figure 1.

To assess MCMC convergence to stationarity, we monitor trace plots and autocorrelation plots of the MCMC iterates for all parameters. Trace plots are obtained via `ts.plot()` for all MCMC iterations, as shown in Figure 2. The red horizontal line is the true value of the parameter, the yellow line represents the posterior mean and the green lines are the lower and upper bounds of the 95% Bayes credible interval. The density plots, via the function `density()`, are provided for each parameter and the differences in the mean coefficients as shown in Figure 3.

An example of a simulation study is now illustrated. For 100 simulated data sets, the code saves the posterior mean, median, standard deviation and the lower and upper bound of each 95% Bayesian interval for each parameter. The means, over the replicated data sets, of these quantities are reported as a table in Figure 4. For counting the frequencies of each estimated delay lag, we provide the frequency table of d by the function `table()`, as shown in the bottom of Figure 4. The average posterior probabilities that $d = 1$ are all very close to 1; the posterior mode of d very accurately estimates the true delay parameter in this case.

US unemployment rate data

For empirical illustration, we consider the monthly U.S. civilian unemployment rate from January 1948 to March 2004. The data set, which consists of 675 monthly observations, is shown in Figure 5. The data is available in Tsay (2005). We take the first difference of the unemployment rates in order to achieve mean stationarity. A partial autocorrelation plot (PACF) of the change of unemployment rate is given in Figure 5. For illustration, we use the same model orders as in Tsay (2005), except for the addition of a 10th lag in regime one. We obtain the fitted SETAR model:

$$y_t = \begin{cases} 0.187y_{t-2} + 0.143y_{t-3} + 0.127y_{t-4} \\ -0.106y_{t-10} - 0.087y_{t-12} + a_t^{(1)} \text{ if } y_{t-3} \le 0.05, \\ 0.312y_{t-2} + 0.223y_{t-3} - 0.234y_{t-12} \\ +a_t^{(2)} \text{ if } y_{t-3} > 0.05, \end{cases}$$

The results are shown in Figure 6. Trace plots and autocorrelograms for after burn-in MCMC iterations are given in Figures 7 and 8. Clearly, MCMC convergence is almost immediate. The parameter estimates are quite reasonable, being similar to the results of Tsay (2005), except the threshold lag, which is set as $d = 1$ by Tsay. Instead, our results suggest that nonlinearity in the differences in the unemployment rate, responds around a positive 0.05 change in the unemployment rate, is at a lag of $d = 3$ months, for this data. This is quite reasonable. The estimated AR coefficients differ between the two regimes, indicating the dynamics of the US unemployment rate are based on the previous quarter's change in rate. It is also clear that the regime variances are significantly different to each other, which can be confirmed by finding a 95% credible interval from the MCMC iterates of the differences between these parameters.

Summary

BAYSTAR provides Bayesian MCMC methods for iterative sampling to provide parameter estimates and inference for the two-regime TAR model. Parsimonious AR specifications between regimes can also be easily employed. A convenient user interface for importing data from a file or specifying true parameter values for simulated data is easy to apply for analysis. Parameter inferences are summarized to an easily readable format. Simultaneously, the checking of convergence can be done by monitoring the MCMC trace plots and autocorrelograms. Simulations illustrated the good performance of the sampling scheme, while a real example illustrated nonlinearity present in the US unemployment rate. In the future we will extend BAYSTAR to more flexible models, such as threshold moving-average (TMA) models and threshold autoregressive moving-average (TARMA) models, which are also frequently used in time series modeling. In addition model and order selection is an important issue for these models. It is interesting to examine the method of the stochastic search variable selection (SSVS) in the R package with BAYSTAR for model order selection in these types of models, e.g. So and Chen (2003).

Acknowledgments

Cathy Chen thanks Professor Kerrie Mengersen for her invitation to appear as keynote speaker and to present this work at the Spring Bayes 2007 workshop. Cathy Chen is supported by the grant: 96-2118-M-002-MY3 from the National Science Council (NSC) of Taiwan and grant 06G27022 from Feng Chia University. The authors would like to thank the editors and anonymous referee for reading and assessing the paper, and especially to thank the referee who made such detailed and careful comments that significantly improved the paper.

Bibliography

P. Brockwell. Beyond Linear Time Series. *Statistica Sinica*, 17:3-7, 2007.

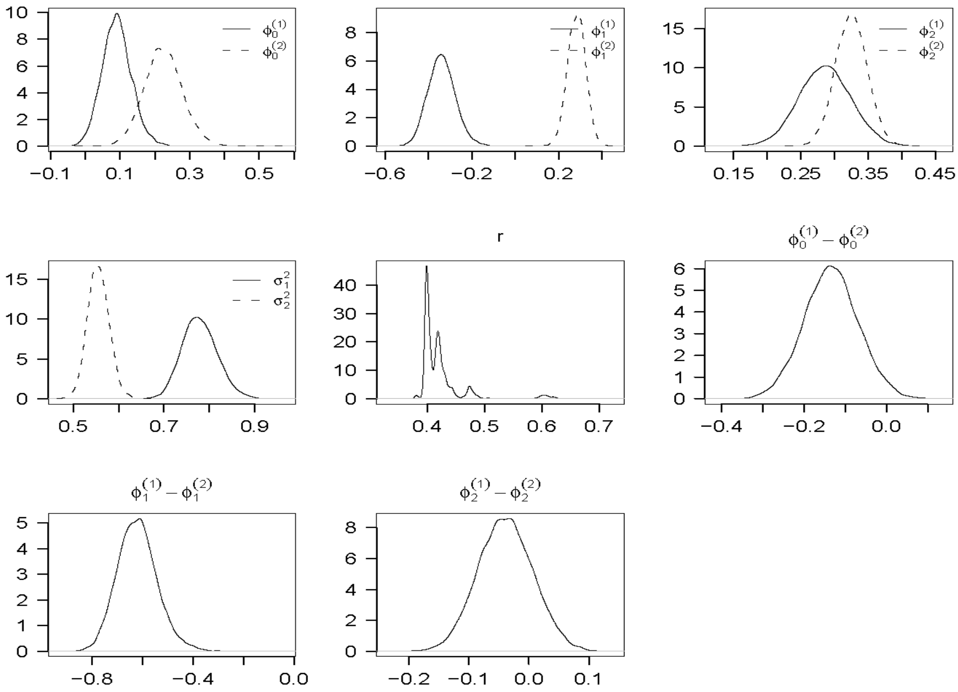

Figure 3: Posterior densities of all parameters and the differences of mean coefficients.

```
              true    mean  median   s.d.   lower   upper
phi0^1      0.1000  0.0917  0.0918 0.0401  0.0129  0.1701
phi1^1     -0.4000 -0.4058 -0.4056 0.0616 -0.5268 -0.2852
phi2^1      0.3000  0.3000  0.3000 0.0417  0.2184  0.3817
phi0^2      0.2000  0.2082  0.2082 0.0509  0.1088  0.3082
phi1^2      0.3000  0.2940  0.2940 0.0387  0.2181  0.3697
phi2^2      0.3000  0.2961  0.2961 0.0226  0.2517  0.3404
sigma1      0.8000  0.7979  0.7966 0.0397  0.7239  0.8796
simga2      0.5000  0.5038  0.5033 0.0209  0.4645  0.5464
r           0.4000  0.3944  0.3948 0.0157  0.3657  0.4247
diff.phi0 -0.1000 -0.1165 -0.1166 0.0644 -0.2425  0.0099
diff.phi1 -0.7000 -0.6997 -0.6995 0.0731 -0.8431 -0.5568
diff.phi2  0.0000  0.0039  0.0040 0.0475 -0.0890  0.0972
mean1       0.0909  0.0841  0.0836 0.0379  0.0116  0.1601
mean2       0.5000  0.4958  0.5024 0.0849  0.3109  0.6434
> table(lag.yt)
lag.yt
  1
100
```

Figure 4: The summary output for all parameters from 100 replications is printed as a table.

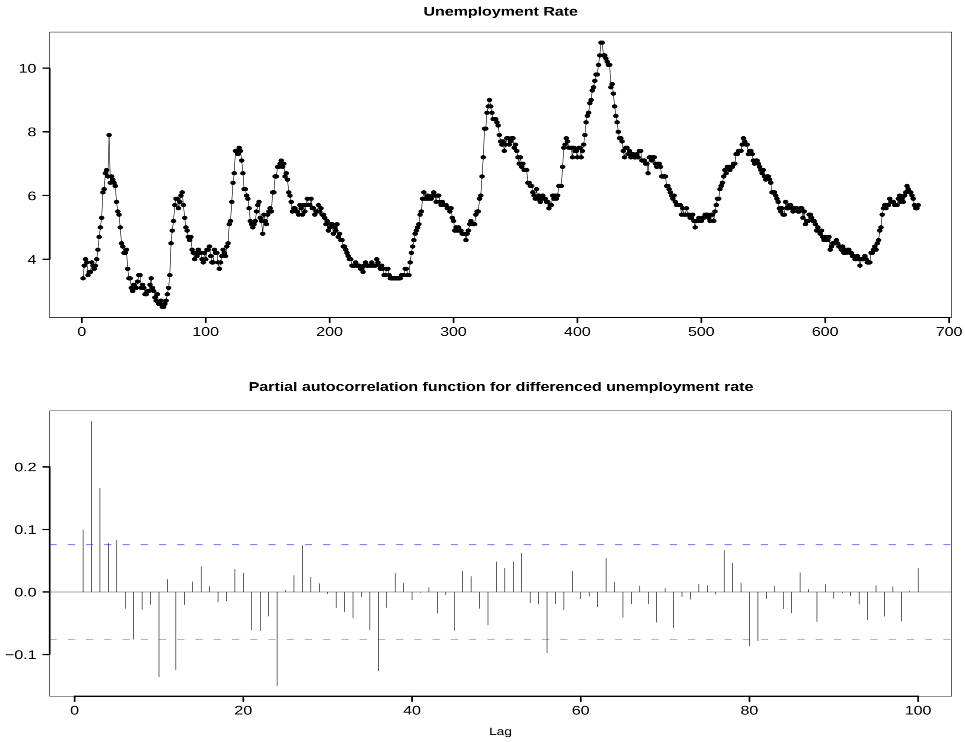

Figure 5: Time series plots of the PACF of the changed unemployment rate.

```
           mean  median   s.d.   lower   upper
phi1.2   0.1874  0.1877 0.0446  0.0993  0.2751
phi1.3   0.1431  0.1435 0.0457  0.0526  0.2338
phi1.4   0.1270  0.1273 0.0447  0.0394  0.2157
phi1.10 -0.1060 -0.1058 0.0400 -0.1855 -0.0275
phi1.12 -0.0875 -0.0880 0.0398 -0.1637 -0.0082
phi2.2   0.3121  0.3124 0.0613  0.1932  0.4349
phi2.3   0.2233  0.2233 0.0594  0.1077  0.3387
phi2.12 -0.2340 -0.2341 0.0766 -0.3837 -0.0839
sigma1   0.0299  0.0298 0.0021  0.0261  0.0342
simga2   0.0588  0.0585 0.0054  0.0492  0.0702
r        0.0503  0.0506 0.0290  0.0027  0.0978
Lag choice :
      1 2    3
Freq 15 0 9985
------------
The highest posterior prob. of lag at :  3
```

Figure 6: The summary output for all parameters of the U.S. unemployment rate is printed as a table.

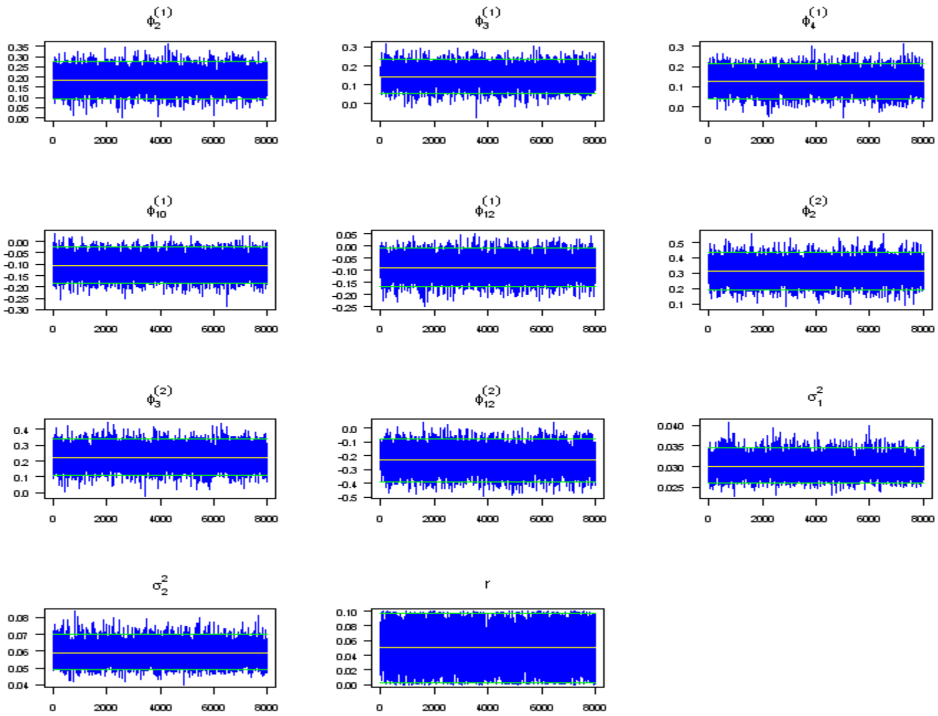

Figure 7: The trace plots of after burn-in MCMC iterations for all parameters.

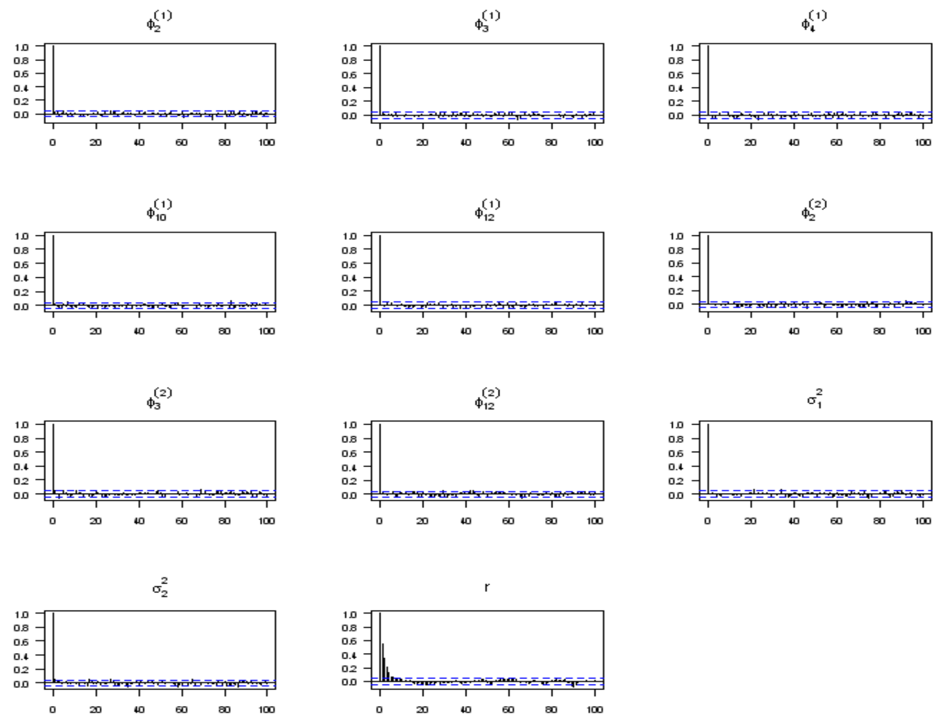

Figure 8: Autocorrelation plots of after burn-in MCMC iterations for all parameters.

C.W.S. Chen. A Bayesian analysis of generalized threshold autoregressive models. *Statistics and Probability Letters*, 40:15–22, 1998.

C.W.S. Chen and J.C. Lee. Bayesian inference of threshold autoregressive models. *J. Time Ser. Anal.*, 16:483–492, 1995.

A. Gelman, G.O. Roberts, and W.R. Gilks. Efficient Metropolis jumping rules. In: *Bayesian Statistics 5* (Edited by J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), 599–607, 1996. Oxford University Press, Oxford.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97-109, 1970.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth and A.H. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087-1091, 1953.

M.K.P. So and C.W.S. Chen. Subset threshold autoregression. *Journal of Forecasting*, 22, :49-66, 2003.

H. Tong. On a Threshold Model in Pattern Recognition and Signal Processing, ed. C. H. Chen, Sijhoff & Noordhoff: Amsterdam, 1978.

H. Tong. Threshold Models in Non-linear Time Series Analysis, Vol. 21 of Lecture Notes in Statistics (K. Krickegerg, ed.). Springer-Verlag, New York, 1983.

H. Tong and K.S. Lim. Threshold autoregression, limit cycles and cyclical data. (with discussion), *J. R. Stat. Soc. Ser. B*, 42:245–292, 1980.

R.S. Tsay. Testing and modeling threshold autoregressive process. *J. Amer. Statist. Assoc.*, 84:231–240, 1989.

R.S. Tsay. Testing and modeling multivariate threshold models. *J. Amer. Statist. Assoc.*, 93:1188–1202, 1998.

R.S. Tsay. Analysis of Financial Time Series, 2nd Edition, John Wiley & Sons, 2005.

Cathy W. S. Chen
Feng Chia University, Taiwan
`chenws@fcu.edu.tw`

Edward M. H. Lin, Feng Chi Liu
Feng Chia University, Taiwan

Richard Gerlach
University of Sydney, Australia
`R.Gerlach@econ.usyd.edu.au`