# EXERCISE: LIKELIHOOD AND MCMC FOR A SIMPLE AR MODEL

Using seasonally adjusted quarterly data for US payroll employment for 1980:I-2009:2, Fit by least squares the second order autoregressive model

$$y_t = \rho_1 y_{t-1} + \rho_2 y_{t-2} + \alpha + \varepsilon_t \,,$$

where $y_t$ is the log of employment at $t$ and $\varepsilon_t$ is assumed i.i.d. $N(0, \sigma^2)$, independent of $y_s$ for $s < t$.

(i) Report the coefficient estimates, the MLE for $\sigma^2$, and the posterior covariance matrix for the coefficient vector $(\rho_1, \rho_2, \alpha)$ under a $1/\sigma$ improper prior.

(ii) As we will learn soon in studying a more general version of this model, if $\mathrm{Var}(y_t)$ and $\mathrm{Cov}(y_t, y_{t-1})$ are constant over time, the roots of the polynomial $1 - \rho_1 L - \rho_2 L^2$ must both be larger than one in absolute value. If they are, then all the $y$'s are jointly normal, with

$$\begin{aligned} \mathrm{Var}(y_t) &= \sigma^2 \frac{1 + r_1 r_2}{(1 - r_1^2)(1 - r_2^2)(1 - r_1 r_2)} \\ \mathrm{Cov}(y_t, y_{t-1}) &= \frac{\rho_1}{1 - \rho_2} \mathrm{Var}(y_t) \\ E[y_t] &= \frac{\alpha}{1 - \rho_1 - \rho_2} \,, \end{aligned}$$

where $r_1$ and $r_2$ are the inverses of the two roots of $1 - \rho_1 L - \rho_2 L^2$, (i.e. the roots of $F^2 - \rho_1 F - \rho_2$). This result allows you to write a likelihood for the entire sample, including the two initial conditions, by multiplying the unconditional joint normal pdf for the initial conditions by the usual likelihood for the rest of the sample conditional on initial conditions. Report an expression for this likelihood and write a program in R, Matlab or the like that evaluates this likelihood, multiplied by the $\sigma^{-1}$ prior, for arbitrary values of $\rho_1$, $\rho_2$, $\alpha$, and $\sigma^2$ with given data. (Note that you will use this formula repeatedly for a fixed data vector, so the "$X'X$" calculation that is required should be done outside the program, for efficiency.)

(iii) In this part of the problem, you construct an MCMC chain to assess how much difference it makes to inference whether one conditions on the initial conditions (as in the OLS estimates) or not. With the initial

*Date*: October 3, 2009.

conditions included in the likelihood, the posterior is no longer in any standard form, so the MCMC is necessary.

In the "independence Metropolis-Hastings" form of an MCMC chain, at each iteration $j$ one draws a new candidate for $\theta^*$ from the same proposal distribution, which does not depend on $\theta_j$. One evaluates the target posterior pdf $\ell_\tau(\theta^*)$ and also the proposal distribution pdf $\ell_\pi(\theta^*)$. One then checks whether

$$\frac{\ell_\tau(\theta^*)\ell_\pi(\theta_j)}{\ell_\tau(\theta_j)\ell_\pi(\theta^*)} > u_j\,,$$

where $u_j$ is a random draw from the $U(0,1)$ (uniform on (0,1)) distribution. If the condition is met, $\theta_{j+1} = \theta^*$. Otherwise $\theta_{j+1} = \theta_j$. In programming this, it is probably best to calculate the logs of both sides of the inequality, rather than the levels, since the pdf values can become extremely large or small in absolute value and might cause an overflow or underflow. Also, your program will need to check at the start that the roots are in the stable region. Otherwise the formulas can produce nonsense or numerical errors that halt the iterations. Unstable coefficients should be treated as having zero likelihood under the target distribution.

Generate an MCMC chain this way, using the Normal-Inverse-Gamma posterior from the OLS estimates as the proposal density and the density that uses initial conditions as the target density. (To draw from this distribution, first draw the Gamma variate, take its inverse as $\sigma^2$, then draw a Gaussian variable from the conditional distribution of the coefficients given $\sigma^2$.) The chain will probably need to be at least 10,000 draws, preferably at least 100,000. One million draws should be plenty, but may take too long unless your code is very efficient or your computer is very fast. My own calculations on a recent vintage computer using R took around 10-30 minutes for one million draws. (I didn't stay around to time it exactly.)

(iv) Check your chain for convergence and accuracy using trace plots and at least one numerical check — e.g., `effectiveSize()` or `geweke.plot()` from the `coda` package.

(v) Construct a scatter plot of $\rho_1$ against $\rho_2$, with the points very small (e.g. `pch="."` in R). If your chain is longer than 10,000, thin it to about 10,000 draws before plotting. On the same graph, show an ellipse for the 95% probability region implied by the posterior conditioned on initial observations. (In R, there is a package called `ellipse` that makes this easy. A Matlab function that is slightly less automatic is at `http://www.mathworks.com/matlabcentral/fileexchange/289.`)

(vi) You will see that the joint distribution of $\rho_1, \rho_2$ shows very strong negative correlation. To get a clearer picture of the difference in the two distributions, plot a similar scatter and ellipse for the transformed variables $\rho_1 + \rho_2, \rho_1 - \rho_2$. These will be much less correlated and will correspond to a much fatter ellipse. Since the transformation is linear, you can transform your draws by postmultiplying the $N \times 2$ matrix of $\rho$ draws by a transformation matrix, and the covariance matrix for the OLS calculations is found by pre and post multiplying the original covariance matrix by the transformation matrix.