SIMPLE MODEL COMPARISON MCMC: ANSWERS

Here is a time series for y_t, $t = 1 \ldots, 10$:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

You are to find the posterior probabilities on two models for this series. One model is that they are i.i.d. draws from a distribution in which $P[y_t = 1] = p$, $P[y_t = 0] = 1 - p$, with a prior on p that is uniform on $[0, 1]$. The other model is that they have been generated from a Markov process in which $P[y_t = 1 \mid y_{t-1} = 1] = p_1$ and $P[y_t = 0 \mid y_{t-1} = 0] = p_2$, with a prior that is uniform over the unit square for p_1, p_2. The initial observation y_1 is under this model drawn from the marginal steady-state distribution of y_t. That is, y_1 is drawn from a distribution in which $P[y_1 = 1] = \bar{p}$ and $\bar{p}p_1 + (1 - \bar{p})(1 - p_2) = \bar{p}$.

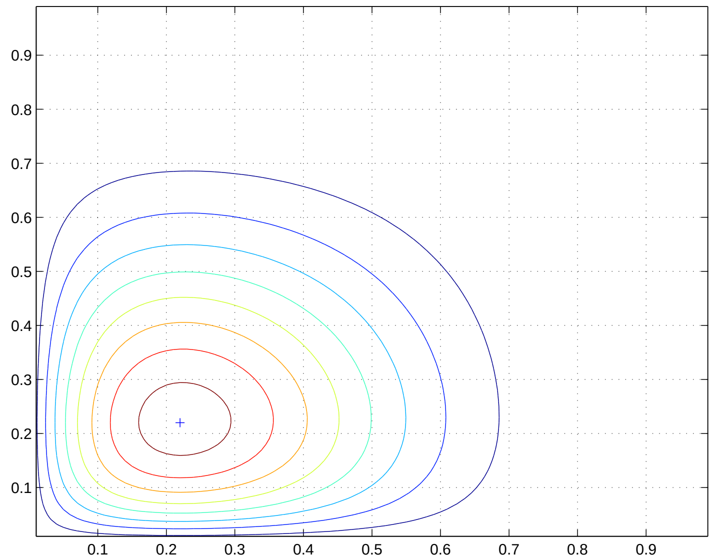
(a) Find the MLE's for p and for p_1, p_2 conditional on each model.

The first model has likelihood function $p^5(1-p)^5$. Since this is symmetric in p and $1-p$, it is obviously maximized at $p = .5$. Fot the second model the likelihood is $\bar{p}(1-p_1)^4(1-p_2)^3 p_1 p_2$. This latter likelihood arises from there being four 1,0 transitions, three 0,1 transitions, and one each of 1,1 and 0,0 transitions. Solving for $\bar{p}$ and writing the whole thing out give us

$$\frac{1 - p_2}{2 - p_1 - p_2} \cdot (1 - p_1)^4 (1 - p_2)^3 p_1 p_2 \,.$$

Since this is a two-dimensional fourth-order polynomial, maximizing it analytically might appear infeasible. Since it is only two-dimensional, though, and on the bounded support of the unit square, a simple grid search will work. A contour plot of the likelihood for the second model, with the peak marked, is below.

©2003 by Christopher A. Sims. This document may be reproduced for educational and research purposes, so long as the copies contain this notice and are retained for personal use or distributed free.



Staring at the algebra a bit would have paid off in this case, though, because it turns out that the likelihood for this sample is symmetric in p_1 and p_2. Once we impose equality on them, the likelihood is easily maximized analytically. The exact maximum is $p_1 = p_2 = 2/9$.

(b) Find the posterior probabilities of the two models, assuming they have equal prior probability, by direct numerical integration, using a grid of, say, 100 points in p-space and 100×100 in (p_1, p_2)-space.

The answer I obtained this way is that the posterior probability of the i.i.d. model is .31073. Matlab code that does the calculation is below

```
p1=.01:.01:.99;
p2=p1;
[P1,P2]=meshgrid(p1,p2);
llh12=-log(2-P1-P2)+4*log(1-P1)+4*log(1-P2)+log(P1)+log(P2);
f12=sum(sum(exp(llh12)))*.0001
> f12 =
>    0.00080023
llh=5*log(p1)+5*log(1-p1);
f=sum(exp(llh))*.01
> f =
>    0.00036075
f/(f12+f)
> ans = 0.31073
```

(c) Using the second-order approximation to the log likelihood at its peak, calculate an approximate posterior odds ratio for the two models.

The peak of the 1-dimensional model at $p = .5$ is $.5^{10} = 0.00097656$. The peak of the two-dimensional model at $p_1 = p_2 = 2/9$ is $(1/(2-4/9)) \cdot (7/9)^8 \cdot (2/9)^2 = 0.0042514$. Minus the second derivative of the log likelihood at the peak for the

1-dimensional model is $10/.5^2 = 40$*, corresponding to a Gaussian distribution with standard error* $1/\sqrt{40} = .15811$*. Minus the second derivative matrix for the log likelihood of the 2-dimensional model has diagonal elements*

$$\frac{1}{(2-2\hat{p})^2} + \frac{4}{(1-\hat{p})^2} + \frac{1}{\hat{p}^2} = 27.276\,,$$

where $\hat{p}$ *is the MLE for both* p_1 *and* p_2 *in this sample. The off diagonal elements are*

$$\frac{1}{(2-2\hat{p})^2} = .41327\,.$$

The inverse of this matrix is the corresponding Gaussian covariance matrix, and its determinant is 0.0013445*. The integrated posteriors based on the local Gaussian approximation are, then*

$$0.00097676 \cdot \sqrt{2\pi} \cdot .15811 = 0.00038711$$
$$0.0042514 \cdot 2\pi \cdot \sqrt{.0013445} = 0.00097947\,.$$

This implies posterior probability of the i.i.d. model is 0.28327*, within 10% of the value found by numerical integration over the grids.*

(d) Calculate posterior odds ratios for the two models by two Monte Carlo methods:
   (i) Importance sampling, using the Gaussian approximations to form the proposal distribution.
   (ii) Metropolis-Hastings sampling, using the Gaussian approximations to form the proposal distributions.

In each case, make at least 5000 draws and present some evidence on whether your algorithm has converged. Also calculate estimates of Monte Carlo standard errors in your estimates of p, p_1, and p_2.

In each case we will use a proposal distribution that puts equal probability on the two models and, conditional on the model, uses

$$N(.5, .15811^2) \qquad \textit{for the i.i.d. model}$$

$$N\left(\begin{bmatrix} .2 \\ .2 \end{bmatrix}, \begin{bmatrix} 0.036671 & -0.00055563 \\ -0.00055563 & 0.036671 \end{bmatrix}\right) \qquad \textit{for the Markov model}$$

Of course we discard draws from the proposal distribution that produce p's outside $[0\ 1]$*, which means that either we retain these draws as "zeros" when evaluating expectations, or we must correct the pdf of the proposal distribution to reflect its truncation. At each draw for the importance-sampling method, we weight by the ratio of the posterior height to the drawn normal pdf value. We don't need to worry about weighting the models themselves, since they are drawn with equal probability. In fact, for the importance sampling, the models need not even be drawn randomly. We can just draw equal numbers from both models. The posterior probabilities of the models are then just estimated as the ratios of the sums or means of drawn weights on the two models.*

For the Metropolis-Hastings method, we draw from the same distribution (except that here it is important to keep the process sequential, with the draws randomly from one model or the other). However now these draws are taken as proposal *draws, and we keep or discard them according to the usual M-H rule. We form*

$$\rho = \frac{\ell(p^*)\phi(p_0)}{\ell(p_0)\phi(p^*)},$$

where p^* is the proposal draw parameter value (which may be a one or two-dimensional vector), p_0 is the parameter value from the previous draw, ℓ is the likelihood value, and ϕ is the normal pdf we are using for our proposal distribution. Then as usual we keep p^* as the new draw if $\rho \geq 1$ and keep it with probability ρ is $\rho < 1$, otherwise repeating p_0. The posterior probability of a model is then estimated as the fraction of draws that are from that model.

My implementation of importance sampling produced a posterior probability of the i.i.d. model of 0.31102, *very close to what was found from numerical integration. The importance weights show only a modest range of variation, so that the sampling standard deviation of the two mean values are only about 2%, and therefore the close match between the importance sampling and direct numerical integration results is not surprising. Code I used to generate the importance sampling results (edited and cleaned up, so it is not absolutely guaranteed to run as shown), is listed below.*

```
d1=randn(2500,1);
s=sqrt(1/40)
d1=.5+d1*s;
lw1=5*log(d1)+5*log(1-d1)+.5*log(2*pi*s^2)+.5*(d1-.5).^2/(s^2);
erff(.5/s) %erff.m, based on the matlab standard erf.m,
           %computes the Gaussian CDF.
1-2*(1-ans)
truncfac1=ans
lw1ok=(d1>0)&(d1<1);
m1w=mean(exp(lw1).*lw1ok)*truncfac1;
d12=randn(2500,2);
d12=d12*chol(s12);
lw12ok=(d12>0)&(d12<1);
lw12ok=lw120k(:,1)&lw12ok(:,2);
truncfac12=sum(lw12ok)/2500;
m12w=sum(exp(lw12).*lw12ok)/2500;
lw12=-log(2-d12(:,1)-d12(:,2))+4*log(1-d12(:,1))+4*log(1-d12(:,2))...
   +log(d12(:,1))+log(d12(:,2))+log(2*pi)+.5*log(det(s12))...
       +.5*sum((((d12-.22222222)/chol(s12)).^2)')';
std(exp(lw12(find(lw12ok))))
m1w/(m1w+m12w)
std(exp(lw12(find(lw12ok))))
std(exp(lw1(find(lw1ok))))
```

Code that implements the Metropolis-Hastings method is displayed below.

```
function [model,p,lr]=mhdraw(oldmodel,oldp,oldlr)
% Metropolis-Hastings draw for exercise.
s1=  [ 0.19149763480010  -0.00290147931515
                  0   0.19147565263452];
f1=chol(s1);
mu=(2/9)*[1,1];
s0=1/sqrt(40);
% draw to determine which model is proposed
model=(rand(1)>.5);
if model %=1, i.e. Markov model
    z=randn(1,2);
    p=z*f1+mu;
    if all((p>0)&(p<1))
        lr=-log(2-sum(p'))+4*log(1-p(1))+4*log(1-p(2))...
            +log(p(1))+log(p(2));
        lr=lr+log(2*pi)+sum(log(diag(f1)))+.5*sum(z.^2);
        if lr>=oldlr
            use=1;
        else
            use=rand(1)<exp(lr-oldlr);
        end
    else
        use=0;
    end
else %model=0, i.e. i.i.d. model
    z=randn(1);
    p=z*s0+.5;
    if p>0 & p<1
        lr=5*log(p)+5*log(1-p)+.5*log(2*pi)+log(s0)+.5*(p-.5)^2/s0^2;
        if lr>=oldlr
            use=1;
        else
            use=rand(1)<exp(lr-oldlr);
        end
    else
        use=0;
    end
end
if ~use
    model=oldmodel;
    p=oldp;
    lr=oldlr;
end
```

```
% interactive code that does the sampling
>for id=1:5000
>[model,p,lr]=mhdraw(model,p,lr);
>mpl(id,:)={model,p,lr};
>end
>mean([mpl{:,1}])
>std([mpl{:,1}])
ans =
   0.69580000000000
ans =
   0.46011379129537
>ans/sqrt(5000)
ans =
   0.00650699163885
> ans*3
% 3 is approximately the square root of the sum of the acf of [mpl{:,1}]
ans =
   0.01952097491654
```

Note that the results of this run make the probability of the i.i.d. model .304, in close agreement with the other results. However, other runs deviated from this one substantially, consistent with the rough Monte Carlo standard error calculated above and reflecting the high serial correlation in the draws.