HIDDEN MARKOV CHAIN MODELS

1. The class of models

We consider a class of models in which we have a parametric model for the conditional distribution of the observation $y_t \,|\, \{y_s,\, s < t\}$ with the pdf $p(y_t \,|\, \theta_t, \{y_s, s < t\})$. The parameter θ_t varies over time and is determined by an unobserved state variable S, so that $\theta_t = \theta(S_t)$. The unobserved state takes on a finite number of values $j = 1, \ldots, m$ and follows a Markov chain, so that

$$P[S_t = i \,|\, S_{t-1} = j] = h_{ij} \,. \tag{1}$$

For such a model it is straightforward, given $H = [h_{ij}]$, a known form for the $\theta(S_t)$ function and a prior distribution for the initial state, to evaluate the pdf of the data (i.e. the likelihood, when it is treated as a function of $\theta(\cdot)$ and H), conditional on the initial observation $y(1)$. In the process we generate filtered estimates of the state, giving $P[S_t = i \,|\, \{y_s, s \le t\}\,, \theta(\cdot), H]$ for all j at each t. The filtered estimates of state probabilities can then be converted by a recursive algorithm to smoothed estimates, giving $P[S_t = i \,|\, \{y_s, s \le T\}\,, \theta(\cdot), H]$, where T is the date of the last observation. A similar recursive algorithm will generate pseudo-random draws from the posterior distribution of the sequence of states $\{S_t, t = 1, \ldots, T\}$ conditional on $\{y_t, t = 1, \ldots, T\}\,, \theta(\cdot), H$.

It is therefore not difficult to program likelihood maximization for these models, and, depending on the form of $\theta(\cdot)$ and on what restrictions there may be on the form of H, also not difficult to program Gibbs sampling from the posterior distribution of the parameters.

2. Uses of and problems with the model

This class of models is obviously attractive because of its tractability. It is also attractive because it easily accommodates discontinuous "regime changes". Models that allow for regime changes but treat them as non-stochastic are usually limited as forecasting tools or for characterizing uncertainty because regime changes in economics usually do have a stochastic, dynamic character — if they have happened once, they can probably happen again, so modeling uncertainty about them is important.

The regime changes do not need to be shifts in mean, or shifts in regression parameters. They can be restricted to shifts in residual variances, so that the model becomes a way of modeling stochastically varying heteroskedasticity, i.e. stochastic volatility. The most common way of modeling stochastic volatility treats residual variance as a continuously distributed random variable with some sort of autoregressive serial dependence. Hidden Markov chain models can easily allow for the possibility of discontinuous sudden jumps in variance, while the more standard model cannot easily do so. Hidden Markov chain models make *all* volatility shifts discontinuous, though, which may not

Date: January 17, 2006.

be appealing. Nonetheless, by using many states and a suitably restricted H matrix, a hidden Markov chain model can approximate a model with continuously distributed, continuously varying heteroskedasticity, as we will explain below.

The model as laid out above allows no dependence of h_{ij} on y. That is, the evolution of the state is entirely exogenous to the stochastic model for determining y_t from the past. In a model of stochastic volatility this restriction is not obviously undesirable, though it certainly would be worth testing. In a model of monetary policy behavior with y a policy-controlled interest rate, this restriction is definitely unappealing. There one thinks of policy "regimes" as shifting in response to the history of inflation and interest rates. The derivations in the next section of filtering, likelihood, smoothing, and sample-drawing for $\{S_t\}$ are all easily generalized to allow dependence of H on y. However, the discussion of drawing artificial samples from the conditional distribution of H given data and other parameters that appears in section 4 would not apply to such a model, and sampling from the posterior on H or on the parameters determining it would be substantially less convenient for models that allowed feedback from y into determination of S.

3. Implementing the calculations for likelihood and for filtered, smoothed and simulated S

Suppose we are given $p(S_t \,|\, \{y_s, s \leq t\}, \theta(\cdot), H)$, the pdf of S_t given data up through time t. (Since this section is entirely concerned with calculations in which $\theta(\cdot)$ and H are held fixed, we will drop them as explicit arguments of pdf's henceforth. Also, we will use p as a generic symbol for a pdf, with the random variables for which p is pdf and the the conditioning information implicit in p's arguments.) We wish to use the observation y_{t+1} to form $p(S_{t+1} \,|\, \{y_s, s \leq t+1\})$. We will henceforth use Y_t to refer to $\{y_s \,|\, s \leq t\}$. We would also like to form $p(y_{t+1} \,|\, Y_t)$, since a product of terms of this form will give us the likelihood function.

Observe that

$$\begin{aligned} p(y_{t+1}, S_{t+1}, S_t \,|\, Y_t) &= p(S_t \,|\, Y_t) \cdot p(S_{t+1} \,|\, Y_t, S_t) \cdot p(y_{t+1} \,|\, Y_t, S_t, S_{t+1}) \\ &= p(S_t \,|\, Y_t) \cdot p(S_{t+1} \,|\, S_t) \cdot p(y_{t+1} \,|\, Y_t, \theta(S_{t+1}))\,. \end{aligned} \tag{2}$$

On the right of the last equality in this expression, the first p is the one we assumed given, the second can be read off from H, and the third is the model for y conditional on parameters that we began with. Thus we know how to evaluate this expression. But then

$$p(y_{t+1} \,|\, Y_t) = \sum_{i,j} p(y_{t+1}, S_{t+1} = i, S_t = j \,|\, Y_t) \tag{3}$$

$$p(S_{t+1} \,|\, Y_{t+1}) = \frac{\sum_j p(y_{t+1}, S_{t+1}, S_t = j \,|\, Y_t)}{p(y_{t+1} \,|\, Y_t)}\,. \tag{4}$$

To form a complete likelihood, we need to know the unconditional distribution of S_1. Usually the observed value of y_1 will in fact be informative about S_1, but calculating the unconditional joint distribution for y_1 and S_1 implied by the dynamic model is difficult, and if the model might be non-stationary, the unconditional distribution might not

even exist. Instead we are treating y_1 as non-stochastic and assuming S_1 is drawn from the unconditional distribution of S alone. This can be found from H as the right-eigenvector of H associated with its unit eigenvalue. This will be a vector $\bar{p}$ satisfying

$$H\bar{p} = \bar{p}\,, \tag{5}$$

which obviously makes $\bar{p}$ a steady-state for the unconditional pdf. It can happen that H has multiple unit-eigenvalues. In that case the Markov process is not ergodic, meaning that when started up from different initial conditions it can converge to different steady-state distributions. (An example: $H = I$, which imply that the process simply remains in whatever state it started in.) In such cases we take $\bar{p}$ to be the average of all the right eigenvectors corresponding to unit roots.

So to start up the recursion, we use $\bar{p}$ in place of the $p(S_1 \,|\, Y_1)$ that the recursion calls for.

The recursion to generate the smoothed distribution of states assumes that we start with $p(S_{t+1} \,|\, Y_T)$ and have available the complete filtering output from the forward recursion, i.e. $p(S_t \,|\, Y_t), t = 1, \ldots, T$). We aim at finding $p(S_t \,|\, Y_T)$, from which we can continue the recursion. We observe

$$\begin{aligned} p(S_{t+1}, S_t \,|\, Y_T) &= p(S_{t+1} \,|\, Y_T) \cdot p(S_t \,|\, S_{t+1}, Y_T) = p(S_{t+1} \,|\, Y_T) \cdot p(S_t \,|\, S_{t+1}, Y_t) \\ &= p(S_{t+1} \,|\, Y_T) \cdot \frac{p(S_{t+1} \,|\, S_t) \cdot p(S_t \,|\, Y_t)}{\sum_j p(S_{t+1} \,|\, S_t = j) \cdot p(S_t = j \,|\, Y_t)}\,. \end{aligned} \tag{6}$$

The second equality follows because y_{t+s} for $s \geq 1$ depends on S_t only through S_{t+1}, so that when S_{t+1} is known, only values of y dated t and earlier provide additional information about S_t. The last equality just applies the formulas relating conditional and joint densities.

The right-hand side of the last equality in (6) involves only factors we have assumed we already know, so we can compute this expression. But then looking at the first left-hand side, it is apparent that if we sum it over all possible values of S_{t+1}, we arrive at our target, $p(S_t \,|\, Y_T)$.

Finally, the recursion to generate a sample path of S_t from its conditional distribution given the data can also be based on (6). Note that the right-hand side of the first equality includes the term $p(S_t \,|\, S_{t+1}, Y_T)$, which is then constructed via the expressions developed in the remaining equalities. Note further that

$$p(S_t \,|\, S_{t+1}, Y_T) = p(S_t \,|\, \{S_{t+s}, s \geq 1\}\,, Y_T)\,, \tag{7}$$

because future S's (like future y's) depend on current S_t only via S_{t+1}. Thus we can implement a backward recursion. Beginning by drawing S_T from the filtered $p(S_T \,|\, Y_T)$, we then draw S_{T-1} from

$$p(S_{T-1} \,|\, S_T, Y_T) = p(S_{T-1} \,|\, S_T, Y_{T-1}) = \frac{p(S_T \,|\, S_{T-1}) \cdot p(S_{T-1} \,|\, Y_{T-1})}{\sum_j p(S_T \,|\, S_{T-1} = j) \cdot p(S_{T-1} = j \,|\, Y_{T-1})}\,, \tag{8}$$

and so on back to $t = 1$.

4. Implementing a full Gibbs chain on all model parameters

The algorithm in the preceding section for making a draw from the distribution of $\{S_t\} \,|\, Y_T$ can form one component of a Gibbs sampling scheme. The other component, of course, would have to be sampling from the posterior pdf of $\theta(\cdot)$ and H conditional on $\{S_t\}$. Whether this is convenient or even feasible depends on the form of $p(y_t \,|\, Y_{t-1}, \theta(\cdot), H)$ and on what restrictions there may be on the form of H.

When H is unrestricted (except for the normalizing identity that requires each column to sum to one), the likelihood depends on it, for fixed $\{S_t\}$, in a straightforward way. It is not likely to be practical to leave H unrestricted except in small models with few states, but the principles here will carry over to some other cases. As a function of the elements h_{ij} of H, the likelihood with the S's fixed is proportional to

$$\bar{p}(S_1) \prod_{i,j} h_{ij}^{n(i,j)} \,, \tag{9}$$

where $n(i,j)$ is the number of dates t, $2 \leq t \leq T$, at which $S_{t-1} = i$ and $S_t = j$. The steady-state probability distribution $\bar{p}$ depends on H via (5). The part of this expression following the $\bar{p}$ term has the form of the product of m independent Dirichlet$(n(1,j)+1, n(2,j)+1, \ldots, n(m,j)+1)$ pdf's, for $j = 1, \ldots, m$. The Dirichlet is a standard distribution and is easy to sample from. The $\bar{p}$ term gives the full expression a non-standard form, but in reasonably large samples the $\bar{p}$ term is likely to be fairly flat relative to the remainder of the expression. Therefore a Metropolis-Hastings sampling scheme, in which the values of H are drawn from the Dirichlet, then an accept/repeat decision is made based on the value of the $\bar{p}$ term under the new and old draw, is likely to be fairly efficient.

This method of sampling from the conditional posterior for H under a flat prior is easily adapted to cases where there are zero restrictions on the H matrix and to non-flat, but Dirichlet, priors. However it may well be attractive to parameterize H more tightly than is possible with zero restrictions alone, and this may make sampling from the H posterior more difficult.

Sampling from the $\theta(\cdot)$ posterior is inevitably model-dependent, so we take that up in our example below.

5. Example: Poor Man's Stochastic Volatility Model

We take $p(y_t \,|\, \theta(\cdot), Y_t)$ from the general model of section 1 to be determined by the equation

$$r_t = \alpha_0 + \alpha_1 r_{t-1} + \varepsilon_t \sigma(S_t) \,, \tag{10}$$

where r is thought of as an interest rate or asset yield and ε is i.i.d. $N(0,1)$. The $\theta(\cdot)$ function is therefore characterized by $m+2$ parameters: The m values of $\sigma_j, j = 1, \ldots, m$ corresponding to the m states, plus α_0 and α_1, which do not vary with the state.

We impose the requirement that $\sigma_j < \sigma_{j+1}$, all j. Since the states differ only in their σ values, we can apply the same permutation to the σ sequence and to the subscripts of the H matrix without changing the model's implications for the behavior of the

observable y's. The restriction that the σ sequence be a monotone function of the state is therefore necessary to avoid redundancy in the parameterization.

Sampling from the conditional posterior of these $m+2$ parameters conditional on H and an $\{S_t\}$ sequence is best accomplished in two steps. First, conditioning on the $\{\sigma_j\}$ values, the likelihood is that of a normal linear regression with known, but time varying variances, i.e. a "weighted least squares" model. The posterior is therefore normal, centered on the least-squares estimates of α_0 and α_1 based on the appropriately weighted data. (Data for a period t in which $S_t = j$ are weighted by $1/\sigma_j$). The covariance matrix of this normal distribution is just the "$(X'X)^{-1}$" matrix for the weighted data. So sampling from this conditional distribution is simple.

With the α's held fixed, the likelihood as a function of the σ's is proportional to

$$\prod_{j=1}^{m} \sigma_j^{-n(j)} e^{-\frac{s_j^2}{2\sigma_j^2}}, \tag{11}$$

where $n(j)$ is the number of occurrences of state j in the $\{S_t\}$ sequence and s_j^2 is the sum of squared residuals, calculated using the fixed values of α, over time periods in which $S_t = j$. The expression (11) is proportional to the product of m inverse-chi-squared pdf's, for with the j'th pdf having $s_j^2/2$ as its inverse-scale parameter and $n(j)-1$ as its shape parameter. It is therefore easy to draw from this distribution. To impose our monotonicity requirement on the σ sequence, we can just discard any draw that violates the requirement. So long as the data contain substantial evidence for time-varying variances, this should result in few discards, though with m large and slight evidence for differences among the σ_j's, the result could be inefficiently many discards.

But we will not want to sample directly from this conditional likelihood in any case. when H implies that a certain state j should be rare, it can easily happen that a draw of the $\{S_t\}$ sequence contains no occurrences of that state. In that case $n(j) = s_j^2 = 0$, and the conditional posterior pdf for σ_j is flat. It therefore cannot be normalized to be a proper pdf and we cannot draw from it. A proper prior pdf on the σ's is therefore necessary so that the conditional posterior remains well-defined for states that fail to occur in a sampled S sequence. A convenient way to introduce such a prior is to add a dummy observation for each state with a squared residual u_0^2, where u_0 is some reasonable guess as to the likely usual size of residuals for the model. This is equivalent to increasing each $n(j)$ count by 1 and each s_j^2 by u_0^2. If the data turn out to imply that some state or states are rare, however, this prior could influence results, so checks (e.g. by varying the choice of u_0) for how sensitive results are to the prior would be important.

6. Software

A set of matlab functions is available on the course website that implement all the calculations described in the preceding sections. Here is a list of the functions, with descriptions of what they do. Note that for the exercise you are required to do for this course, only the last two of these programs (plus a function minimizer) is needed. The

others are described here so that you will understand how the calculations are being done and so that you could modify them yourself to estimate some model other than the model of section 5 (referred to henceforth as the PMSV model).

[lh,sp]=mclh(y,x,sig,b,H): Generates likelihood and filtered state probabilities for given values of the parameters.

y: $T \times 1$ vector of left-hand-side variable data.

x: $T \times k$ matrix of right-hand-side variable data (lagged r and a column of ones for the PMSV model).

sig: The $1 \times m$ vector of residual standard deviations, as a function of the state

b: The $k \times m$ matrix of coefficient vectors for the m states (all columns identical for the PMSV model).

H: The $m \times m$ matrix of state transition probabilities.

lh: The $T \times 1$ vector of multiplicative likelihood increments. The product of these is the likelihood, the sum of their logs the log likelihood.

sp: The $T \times m$ matrix of filtered state probabilities. Each row is the conditional probability distribution for S_t given Y_t.

smsp=mcsmooth(H,sp): Generates smoothed state probabilities from the filtered probabilities and the transition probability matrix.

smsp: The $T \times m$ matrix of smoothed state probabilities. Each row is the conditional probability distribution for S_t given Y_T.

spdraw=mcdraw(H,sp): Generates a draw from the posterior on $\{S_t\}$ from the transition matrix and the filtered state probabilities.

spdraw: A $T \times 1$ vector representing a draw from the posterior pdf of the $\{S_t\}$ sequence.

H=mcHdraw(spdraw,oldH): Generates a Metropolis-Hastings draw from the conditional posterior on the transition matrix H from a given draw on $\{S_t\}$ and the previous value of H.

H: A Metropolis-Hastings draw from the conditional posterior on H.

OldH: The previous draw's value for H, which the Metropolis-Hastings algorithm may decide to repeat.

b=mcbdraw(y,x,sig,spdraw): Generates a draw from the conditional posterior of the regression parameter vector (α in the PMSV model).

b: A $k \times 1$ vector representing a draw from the posterior probability over the coefficient vector. Assumes that only σ, not the regression coefficients, varies with the state.

sig=mcsdraw(y,x,b,sdraw): Generates a draw from the conditional posterior on σ. Note that this routine incorporates a prior with a particular value of u_0 and may need adjustment and/or sensitivity analysis.

[bdraw,sigdraw,pdraw,spsum,spssq] =mcexGibbs(y,x,b0,sig0,p0,nit): Generates a set of Monte-Carlo draws from the joint posterior pdf on all the parameters of the PMSV model.

b0: A starting value for the $k \times 1$ coefficient vector.

sig0: A starting value for the $1 \times m$ vector of state-specific residual standard deviations.

`p0`: A starting value for the diagonal of the H matrix. This assumes that $m = 2$, so that these two values determine the entire H matrix.

`nit`: The number of Monte Carlo draws. On a 500Mhz Pentium III with 128k, 1000 draws for the PMSV model complete in a few minutes. 10000 take over an hour.

`bdraw`: `nit`$\times k$ matrix of draws on the regression coefficients.

`sigdraw`: `nit`$\times m$ matrix of draws on the $\{\sigma_j\}$ vector.

`pdraw`: `nit`$\times m$ matrix of draws of the diagonal of H. Here again, $m = 2$ is assumed.

`spsum`: $T \times 1$ vector of the sums of the draws of the smoothed estimate of the probability of state 1.

`spssq`: $T \times 1$ vector of the sums of squares of the draws of the probability of state 1. From `spsum` and `spssq` the posterior means and standard errors of the state probabilities at each date can be constructed.

`lh=mcexlh(z,y,x)`: Returns minus the log likelihood for the PMSV model, with all the parameters packed in to a single vector. This is the way the likelihood evaluation must be set up if the function is to be used as input to a function minimization routine, like `csminsel.m` or the similar routines packaged with matlab. Note that the program does not weight by the proper prior on σ that is used in the Gibbs sampling program, so if it is used for likelihood maximization it does not find exactly the posterior mode of the distribution from which the Gibbs sampling program generates draws.

`z`: This is a 6×1 vector with elements $\alpha_0, \alpha_1, \sigma_1, \sigma_2, h_{11}, h_{22}$.

7. Exercise (not required in the 2005-6 version of the course)

(a) Using the monthly data on the Federal Funds rate available on the course web site, use `mcexlh` together with `csminwel` or another function minimization routine to find the maximum likelihood estimate of the PMSV model parameters.

(b) Use mcexGibbs to generate draws from the posterior of the PMSV model for the Fed Funds data. Use at least three starting points and at least 1000 Monte Carlo draws from each staring point. Assess convergence by plotting the draws for the parameters as functions of iteration number and by computing "effective sample size" based on between and within variances.

(c) Compare the maximum likelihood to the posterior mean values of the regression coefficients and σ. Are the differences large relative to the posterior uncertainty (posterior standard deviations or inter-quantile ranges)?

(d) Construct a histogram for each of the 6 model parameters (matlab's `hist` command does this automatically) from the Monte Carlo data.

(e) Without estimating a new model, assess whether this model has adequately captured the time variation in residual variances. (E.g., you might start by plotting residuals standardized by the model's posterior mean value of $\sigma(S_t)$ for the value of S_t with highest posterior probability at each date.)