Bank of Canada Working Papers

1995

95-1	Deriving Agents' Inflation Forecasts from the Term Structure of Interest Rates	C. Ragan
95-2	Estimating and Projecting Potential Output Using Structural VAR Methodology: The Case of the Mexican Economy	A. DeSerres, A. Guay and P. St-Amant
95-3	Empirical Evidence on the Cost of Adjustment and Dynamic Labour Demand	R. A. Amano
95-4	Government Debt and Deficits in Canada: A Macro Simulation Analysis	T. Macklem, D. Rose and R. Tetlow
95-5	Changes in the Inflation Process in Canada: Evidence and Implications	D. Hostland
95-6	Inflation, Learning and Monetary Policy Regimes in the G-7 Economies	N. Ricketts and D. Rose
95-7	Analytical Derivatives for Markov Switching Models	J. Gable, S. van Norden and R. Vigfusson

1994

(Earlier 1994 papers, not listed here, are also available.)

94-5	Exchange Rate Volatility and Trade: A Survey	A. Côté
94-6	The Dynamic Behaviour of Canadian Imports and the Linear-Quadratic Model: Evidence Based on the Euler Equation	R. A. Amano and T. S. Wirjanto
94-7	L'endettement du secteur privé au Canada : un examen macroéconomique	J.-F. Fillion
94-8	An Empirical Investigation into Government Spending and Private Sector Behaviour	R. A. Amano and T. S. Wirjanto
94-9	Symétrie des chocs touchant les régions canadiennes et choix d'un régime de change	A. DeSerres and R. Lalonde
94-10	Les provinces canadiennes et la convergence : une évaluation empirique	M. Lefebvre
94-11	The Causes of Unemployment in Canada: A Review of the Evidence	S. S. Poloz
94-12	Searching for the Liquidity Effect in Canada	B. Fung and R. Gupta

Single copies of Bank of Canada papers may be obtained from

Publications Distribution
Bank of Canada
234 Wellington Street
Ottawa, Ontario K1A 0G9

The papers are also available by anonymous FTP to the following address, in the subdirectory /pub/publications:

ftp.bank-banque-canada.ca

```
@Kim's Smoothing Algorithm Kim(1994)@
x = pxp1[1:nrows-1,.].*(p[2:nrows,.]./pxa1[2:nrows,.] - (1-p[2:nrows,.])./(1-
pxa1[2:nrows,.]));
x = pxp1[nrows,.]|rev(x);
z = pxp1[1:nrows-1,.].*(1-p[2:nrows,.])./(1-pxa1[2:nrows,.]);
z = zeros(1,1)|rev(z);
pxpkim = rev(recsercp(x,z));
retp(pxpkim,pxa1,pxp1);

endp;
```

6.0 References

Diebold, Francis X. Joon Haeng Lee and Gretchen C. Weinbach. 1994. “Regime Switching with Time Varying Transition Probabilities.” in C. Hargreaves *Nonstationary Time Series Analysis and Cointegration* Oxford, Oxford University Press.

Durland, J. Michael. and Thomas H. McCurdy. 1994. “Duration Dependent Transitions in a Markov Model of US GNP Growth.” *Journal of Business & Economic Statistics.* 12(July):279-88

Goldfeld,S.M. and R.E. Quandt. 1973. “A Markov model for switching regressions.” *Journal of Econometrics.* 1:3-16

Hamilton, James D. 1989. “A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle.” *Econometrica* 57:357-84

Hamilton, James D. 1992. “Estimation, Inference, and Forecasting of Time Series Subject to Changes in Regime.” in C.R. Rao and G.S. Maddala. *Handbook of Statistics.* Volume 10

Hamilton, James D. 1994. *Time Series Analysis*. Princeton: Princeton University Press

Kim, Chaing-Jin. 1994. “Dynamic Linear Models with Markov-Switching.” *Journal of Econometrics.* 60:1-22

Lam, Pok-sang. 1990. “The Hamilton Model with a General Autoregressive Component Estimation and Comparison with Other Models of Economic Time Series.” *Journal of Monetary Economics.* 26:409-32

```
@ Establish values of parameters @
  alpha1 = th[1:vei[1,1],1];
  alpha2 = th[1+vei[1,1]:vei[1,1]+vei[2,1],1];
  pth = th[1+sumc(vei[1:nstat,1]):sumc(vei[1:nstat,1])+vei[3,1],1];
  qth = th[1+sumc(vei[1:nstat+1,1]):sumc(vei[1:nstat+1,1])+vei[4,1],1];
  sig1 = abs(th[sumc(vei)+1,1]);
  if ix == 0;
    sig2 = sig1;
  else;
    sig2 = abs(th[sumc(vei)+2,1]);
  endif;
nrows=rows(y);
  lf = zeros(nrows,1);

@ Set the p and q for each period @


 p = cdfn(y[.,terms[nstat+1,1]:terms[nstat+1,2]]*pth);
  q = cdfn(y[.,terms[nstat+2,1]:terms[nstat+2,2]]*qth); /**/


@ Select the correct value of rho (prob of state 1 at time 0) @
  if iy == 0;             rho = (1-meanc(q))/(2-meanc(p)-meanc(q));
  elseif iy == 1;         rho = cdfn(th[sumc(vei)+2+ix,1]);
  elseif iy == 2;         rho = 1.0;
  elseif iy == 3;         rho = 0;
  endif;

  pxp2 = 1-rho;
  pxp1 = rho;

@ Set PDF's of y for each regime @
  qq1 = pdfn((y[.,1]-y[.,terms[1,1]:terms[1,2]]*alpha1)/sig1)/sig1;
  qq2 = pdfn((y[.,1]-y[.,terms[2,1]:terms[2,2]]*alpha2)/sig2)/sig2;
    @ This allows for underflows in the above expression. @
    if ndpchk(3);   ndpclex;    endif;

it = 1;     pxp1=zeros(nrows,1);    fit=zeros(nrows,1);     pxa1=zeros(nrows,1);

    do until it > nrows;
      if it == 1;                  @ Step 1 : Ex Ante probability of State 1 @
        pxa1[it,1] = (1-q[it,1])*(1-rho) + p[it,1]*rho;
      else;
        pxa1[it,1] = (1-q[it,1])*(1-pxp1[it-1,1]) + p[it,1]*pxp1[it-1,1];
      endif;
     pxp1[it,1] = pxa1[it,1]*qq1[it,1]; @Step 2:Ex Post joint density of State 1
@
                                   @ Step 3 : Likelihood Function @
      fit[it,1] = pxp1[it,1]+(1-pxa1[it,1])*qq2[it,1];
                                   @ Step 4 : Ex Post probability of State 1 @
      pxp1[it,1] = pxp1[it,1]/fit[it,1];
    it = it+1;
    endo;
```

```
  q = cdfn(y[.,terms[nstat+2,1]:terms[nstat+2,2]]*qth); /**/


@ Select the correct value of rho (prob of state 1 at time 0) @
  if iy == 0;             rho = (1-meanc(q))/(2-meanc(p)-meanc(q));
  elseif iy == 1;         rho = cdfn(th[sumc(vei)+2+ix,1]);
  elseif iy == 2;         rho = 1.0;
  elseif iy == 3;         rho = 0;
  endif;

  pxp2 = 1-rho;
  pxp1 = rho;

@ Set PDF's of y for each regime @
  qq1 = pdfn((y[.,1]-y[.,terms[1,1]:terms[1,2]]*alpha1)/sig1)/sig1;
  qq2 = pdfn((y[.,1]-y[.,terms[2,1]:terms[2,2]]*alpha2)/sig2)/sig2;
    @ This allows for underflows in the above expression. @
    if ndpchk(3);   ndpclex;    endif;
/*
    NOTE: The qq's in the above expressions differs from those in the Hamilton
    likelihood function LIKEPROC.  First, the division by sig2 and sig1 is
    moved up from the lines setting pxp2 and pxp1, below.  Second, they are
    smaller by a factor of sqrt(2*pi); this is required to make it a true
    log-likelihood.  To correct; Hamilton's llf - llf swmkv = (n/2)*ln(2*pi).
*/


it = 1;     pxp1=zeros(nrows,1);    fit=zeros(nrows,1);     pxa1=zeros(nrows,1);

    do until it > nrows;
      if it == 1;                   @ Step 1 : Ex Ante probability of State 1 @
        pxa1[it,1] = (1-q[it,1])*(1-rho) + p[it,1]*rho;
      else;
        pxa1[it,1] = (1-q[it,1])*(1-pxp1[it-1,1]) + p[it,1]*pxp1[it-1,1];
      endif;
      pxp1[it,1] = pxa1[it,1]*qq1[it,1]; @ Step 2 : Ex Post joint density of
State 1 @
                                    @ Step 3 : Likelihood Function @
      fit[it,1] = pxp1[it,1]+(1-pxa1[it,1])*qq2[it,1];
                                    @ Step 4 : Ex Post probability of State 1 @
      pxp1[it,1] = pxp1[it,1]/fit[it,1];
    it = it+1;
    endo;
retp(ln(fit));
endp;
```

5.3 Kim's Smoother

```
proc(3) =kimsmth(th,y);


 local  alpha1,alpha2,p,q,sig2,sig1,rho,pxp2,pxp1,
          pxa1,pxpkim,fit,it,qq1,qq2,lf,pth,qth,x,z,nrows;
```

```
pcr=(pkim[1,1]*rho)./pxa[1,1];
ppcr=(pkim[1,2]*rho)./(1-pxa[1,1]);
pcr=pcr-ppcr;

pcr=pcr.*y[1,terms[nstat+1,1]:terms[nstat+1,2]].*pdfn(y[1,terms[nstat+1,1]:terms
[nstat+1,2]]*beta[pstart:pfin,1]);

endif;

if iy /= 2;
qcr= (pkim[1,2]*(1-rho))./(1-pxa[1,1]);
qqcr= (pkim[1,1]*(1-rho) )./pxa[1,1];
qcr=qcr-qqcr;

qcr=qcr.*y[1,terms[nstat+2,1]:terms[nstat+2,2]].*pdfn(y[1,terms[nstat+2,1]:terms
[nstat+2,2]]*beta[qstart:qfin,1]);
endif;
p=pcr|p;
q=qcr|q;

p[.,1]=p[.,1]+ultp';
q[.,1]=q[.,1]+ultq';

angrd[.,pstart:pfin]=p;
angrd[.,qstart:qfin]=q;

retp(angrd');
endp;
```

5.2 The Likelihood Function

```
proc(1) =swmkv(th,y);

local  alpha1,alpha2,p,q,sig2,sig1,rho,pxp2,pxp1,
          pxa1,pxpkim,fit,it,qq1,qq2,lf,pth,qth,x,z,nrows;

@ Establish values of parameters @
  alpha1 = th[1:vei[1,1],1];
  alpha2 = th[1+vei[1,1]:vei[1,1]+vei[2,1],1];
  pth = th[1+sumc(vei[1:nstat,1]):sumc(vei[1:nstat,1])+vei[3,1],1];
  qth = th[1+sumc(vei[1:nstat+1,1]):sumc(vei[1:nstat+1,1])+vei[4,1],1];
  sig1 = abs(th[sumc(vei)+1,1]);
  if ix == 0;
    sig2 = sig1;
  else;
    sig2 = abs(th[sumc(vei)+2,1]);
  endif;
nrows=rows(y);
  lf = zeros(nrows,1);

@ Set the p and q for each period @


 p = cdfn(y[.,terms[nstat+1,1]:terms[nstat+1,2]]*pth);
```

```
p =(pkim[2:nobs,1].*pxp[1:nobs-1,1])./pxa[2:nobs,1];
pp = (pkim[2:nobs,2].*pxp[1:nobs-1,1])./(1-pxa[2:nobs,1]);
q = (pkim[2:nobs,2].*(1-pxp[1:nobs-1,1]))./(1-pxa[2:nobs,1]);
qq = (pkim[2:nobs,1].*(1-pxp[1:nobs-1,1]))./(pxa[2:nobs,1]);
p=p-pp;
q=q-qq;

p=p.*y[2:nobs,terms[nstat+1,1]:terms[nstat+1,2]].*pdfn(y[2:nobs,terms[nstat+1,1]
:terms[nstat+1,2]]*beta[pstart:pfin,1]);

q=q.*y[2:nobs,terms[nstat+2,1]:terms[nstat+2,2]].*pdfn(y[2:nobs,terms[nstat+2,1]
:terms[nstat+2,2]]*beta[qstart:qfin,1]);

pcr=0;
qcr=0;
ultp=0;
ultq=0;

 if iy == 0; rho = (1-meanc(s2s2))/(2-meanc(s1s1)-meanc(s2s2));

pkimso1=rho*(s1s1[1,1]*pkim[1,1]/pxa[1,1]+(1-s1s1[1,1])*(1-pkim[1,1])/(1-
pxa[1,1]));
pkimso2= 1- pkimso1;

ultp=(pkimso1-rho)/(1-rho);
ultp=ultp/(nobs*(2-meanc(s1s1)-meanc(s2s2)));
ultp=ultp*sumc(
pdfn(y[1:nobs,terms[nstat+1,1]:terms[nstat+1,2]]*beta[pstart:pfin,1])
.*y[1:nobs,terms[nstat+1,1]:terms[nstat+1,2]]);

ultq=(pkimso2-(1-rho))/(rho);
ultq=ultq/(nobs*(2-meanc(s1s1)-meanc(s2s2)));
ultq=ultq*sumc(
pdfn(y[1:nobs,terms[nstat+2,1]:terms[nstat+2,2]]*beta[qstart:qfin,1])
.*y[1:nobs,terms[nstat+2,1]:terms[nstat+2,2]]);

elseif iy == 1; rho=cdfn(beta[numvar+2+ix*(nstat-1),1]);

pkimso1 = rho*(s1s1[1,1]*pkim[1,1]/pxa[1,1]+(1-s1s1[1,1])*(1-pkim[1,1])/(1-
pxa[1,1]));
pkimso2 = 1-pkimso1;

angrd[numvar+2+ix*(nstat-1),1]=pkimso1*pdfn(beta[numvar+2+ix*(nstat-1),1])/
(cdfn(beta[numvar+2+ix*(nstat-1),1])*(1-cdfn(beta[numvar+2+ix*(nstat-1),1])));
angrd[numvar+2+ix*(nstat-1),1]= angrd[numvar+2+ix*(nstat-1),1]-
pdfn(beta[numvar+2+ix*(nstat-1),1])/(1-cdfn(beta[numvar+2+ix*(nstat-1),1]));

@For the next two cases the derivative with respect to intial conditions is zero@

elseif iy == 2; rho = 1.0;
elseif iy == 3; rho = 0;
endif;

if iy /= 3;
```

```
local pxa,pxp,p,pp,qq,q, pkim,rho,pcr,ppcr,qcr,qqcr,begz,
endz,numvar,sigma,errz,angrd, i,nobs,ultp,ultq,pkimso1,
pkimso2,pstart,qstart,pfin,qfin,s1s1,s2s2;

nobs=rows(y);
endz=0;
i=0;
numvar=sumc(vei);
errz=zeros(nobs,nstat);
angrd=zeros(nobs,rows(beta));
{pkim,pxa,pxp}=kimsmth(beta,y);
pkim=pkim~(1-pkim);
@The above was done so it could be handled in a loop. With the n-state smoother
it will not be needed.@

do while i<nstat;
i=i+1;
begz =1+endz;
endz=begz+ vei[i,1]-1;

if ix==1;
sigma=beta[numvar+i,1];
else;
sigma= beta[numvar+1,1];
endif;

errz[.,i]=y[.,1]-(y[.,terms[i,1]:terms[i,2]]*beta[begz:endz,1]);
errz[.,i]=errz[.,i]/(sigma);
angrd[.,begz:endz]=(errz[.,i].*y[.,terms[i,1]:terms[i,2]]).*pkim[.,i]/sigma;

@Sigma gradients when they differ across states@
if ix == 1;
angrd[.,numvar+i]=(errz[.,i]^2-1).*pkim[.,i] /sigma;
endif;

endo;

@Calculate the sigma gradients if ix = 0@
if ix == 0;
angrd[.,numvar+1]=sumc(((errz[.,.]^2-ones(rows(errz),nstat)).*pkim[.,.])' /
sigma);
endif;



@Calculate the prob gradients. The modification to n-states should be possible. @

pstart = sumc(vei[1:nstat,1])+1;
pfin = pstart +vei[nstat+1,1]-1;
qstart = pfin+1;
qfin = qstart + vei[nstat+2,1]-1;


s1s1 = cdfn(y[.,terms[nstat+1,1]:terms[nstat+1,2]]*beta[pstart:pfin,1]);
s2s2= cdfn(y[.,terms[nstat+2,1]:terms[nstat+2,2]]*beta[qstart:qfin,1]);
```

derivative of the likelihood function with respect to γ_{11} and γ_{22} differs from (EQ 39) by the amount shown in (EQ 41) and (EQ 42).

5.0 Program Appendix

The procedure ANGRAD.G is written for use with the matrix programming language Gauss. The procedure conforms with the requirements of Gauss's maximum likelihood estimation routines Maxlik. (In particular, the procedure returns the matrix of the derivative evaluated at each observation rather than the sum as presented in the paper.)

ANGRAD.G requires the global variables *nstat, vei, ix, iy, terms.* The variable *nstat* is the number of states. At present *nstat* must be equal to two. The variable *vei* is an $nstat^2$ by 1 vector. The first *nstat* entries specify the number of variables in the corresponding level equation. The next *nstat (nstat-1)* specify the number of variables in the corresponding transition equation. The variable *ix* controls whether variances differ across states or are equal. The variable *iy* sets the starting condition for the probability of being in the different states at time 0. TABLE 5. gives the possible settings for *ix* and *iy.*

TABLE 5. Settings for *ix* and *iy*

Value	*ix*	*iy*
0	Variances Equal Across States	rho=$\frac{1-q}{2-p-q}$
1	Variances Differ Across States	rho is a parameter
2		rho =0
3		rho=1

Each row of *terms,* an $nstat^2$ by two matrix, specifies what data is used by a particular equation. The first *nstat* rows are for the level equations. The rest are for the transition equations. The first column of *terms* specifies the first column of data used in an equation. The second specifies the last. The dependent variable is assumed to occupy the first column of the data matrix. The elements in *terms* may be specified such that the same variables are used in more than one equation.

The analytical gradients and likelihood function currently can be used only for a two-state system. The code has been written such that the change to an n-state system, while non-trivial, is certainly feasible. The code for the derivatives of (EQ 1) is already capable of handling n-states. The global variables *vei*, *terms*, and *nstat* should facilitate the transition to the n-state system.

5.1 Angrad.g

```
@Analytical Gradients for Two-State Markov process@

proc (1)=angrad(beta,y);
```

(C) We might instead restrict the time 0 densities to be equal to the mean densities over the period $t \in [1, T]$. To date, we only have a solution for the analytical derivatives in the $K = 2$ case. For γ_{11}, it is greater than that in (EQ 39) by the amount

$$p(s_0 = 1 \mid Y_T, \lambda) \cdot \left(\frac{\partial}{\partial \gamma_{11}} \log p(s = 1 \mid \rho) \right) + (1 - p(s_0 = 1 \mid Y_T, \lambda)) \cdot \left(\frac{\partial}{\partial \gamma_{11}} \log p(s = 2 \mid \rho) \right)$$

$$= \frac{p(s_0 = 1 \mid Y_T, \lambda)}{p(s = 1 \mid \rho)} \cdot \frac{\partial}{\partial \gamma_{11}} p(s = 1 \mid \rho) + \frac{1 - p(s_0 = 1 \mid Y_T, \lambda)}{p(s = 2 \mid \rho)} \cdot \frac{\partial}{\partial \gamma_{11}} (1 - p(s = 1\,(2) \mid \rho))$$

$$= \frac{p(s_0 = 1 \mid Y_T, \lambda)}{p(s = 1 \mid \rho)} \frac{p(s = 1 \mid \rho)}{2 - \bar{p}_{11} - \bar{p}_{22}} \cdot \frac{\partial \bar{p}_{11}}{\partial \gamma_{11}} + \frac{1 - p(s_0 = 1 \mid Y_T, \lambda)}{p(s = 2 \mid \rho)} \cdot \left(\frac{-p(s = 1 \mid \rho)}{2 - \bar{p}_{11} - \bar{p}_{22}} \right) \frac{\partial \bar{p}_{11}}{\partial \gamma_{11}}$$

$$= \left(\frac{p(s_0 = 1 \mid Y_T, \lambda)}{p(s = 1 \mid \rho)} - \frac{(1 - p(s_0 = 1 \mid Y_T, \lambda))}{p(s = 2 \mid \rho)} \right) \left(\frac{p(s = 1 \mid \rho)}{2 - \bar{p}_{11} - \bar{p}_{22}} \right) \cdot \frac{\partial \bar{p}_{11}}{\partial \gamma_{11}}$$

$$= \left(\frac{p(s_0 = 1 \mid Y_T, \lambda) - p(s = 1 \mid \rho)}{p(s = 1 \mid \rho) p(s = 2 \mid \rho)} \right) \left(\frac{p(s = 1 \mid \rho)}{2 - \bar{p}_{11} - \bar{p}_{22}} \right) \cdot \frac{\partial \bar{p}_{11}}{\partial \gamma_{11}}$$

$$= \left(\frac{p(s_0 = 1 \mid Y_T, \lambda) - p(s = 1 \mid \rho)}{(2 - \bar{p}_{11} - \bar{p}_{22})\ (1 - p(s = 1 \mid \rho))} \right) \cdot \frac{\sum_{t=1}^{T} \phi(z_t \cdot \gamma_{11}) \cdot z_t}{T}$$

$$= \left(\frac{p(s_0 = 1 \mid Y_T, \lambda) - p(s = 1 \mid \rho)}{(1 - \bar{p}_{11})} \right) \cdot \frac{\sum_{t=1}^{T} \phi(z_t \cdot \gamma_{11}) \cdot z_t}{T} \qquad \text{(EQ 41)}$$

By similar reasoning we can find that for γ_{22}, it is greater than that in (EQ 39) by the amount

$$\left(\frac{p(s_0 = 2 \mid Y_T, \lambda) - p(s = 2 \mid \rho)}{(1 - \bar{p}_{22})} \right) \cdot \frac{\sum_{t=1}^{T} \phi(z_t \cdot \gamma_{22}) \cdot z_t}{T} \qquad \text{(EQ 42)}$$

Note that both (EQ 41) and (EQ 42) require that we evaluate an expression of the form $p(s_0 \mid Y_T, \lambda)$, whose formula is found in (EQ 37). To recap, this means that in case (C) the

Finally, we need to calculate $p(s_0|Y_T, \lambda)$. It follows from Hamilton's (1994) presentation of Kim's smoother that

$$p(s_0|Y_T, \lambda) = p(s_0|Y_0, \lambda) \sum_{s_1} p(s_1|s_0) \frac{p(s_1|Y_T, \lambda)}{p(s_1|Y_1, \lambda)}$$

$$= p(s_0|\lambda) \sum_{s_1} p(s_1|s_0) \frac{p(s_1|Y_T, \lambda)}{p(s_1|Y_1, \lambda)} \qquad \text{(EQ 37)}$$

The terms $p(s_1|Y_T, \lambda)$ and $p(s_1|Y_1, \lambda)$ are available from the calculation of Kim's smoother. As Y_0 provides no information on future values of Y, we can substitute $p(s_0|\lambda)$ for $p(s_0|Y_0, \lambda)$.

Having found the expressions for these various components, we now need only combine them in (EQ 16), keeping in mind how the filter is initialized and the restriction imposed by (EQ 29). The simplest way to impose such a restriction is to assume that

$$p(s_t = K|s_{t-1} = j) = 1 - \sum_{i=1}^{K-1} p(s_t = i|s_{t-1} = j) \qquad \text{(EQ 38)}$$

We can now consider the three different ways of initializing the filter.

(A) If we simply fix the probabilities of being in state *i* at time 0 and treat them as given, then

$$\frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{t=1}^{T} \frac{p(s_{t-1}|\lambda, Y_t)\, p(s_t|Y_T, \lambda)}{p(s_t|\lambda, Y_t)} \cdot \phi(z_t \cdot \gamma_{s_t s_{t-1}}) \cdot z_t, \text{ where}$$

$$\{s_t, s_{t-1} | \rho_j \in \gamma_{s_t s_{t-1}}\}. \qquad \text{(EQ 39)}$$

(B) If instead we treat them as *K-1* additional parameters (ι) over which we must optimize, then $\frac{\partial}{\partial \rho_j} p(Y_T|\lambda)$ is given by (EQ 39) for $\rho_j \notin \iota$, and by

$$\frac{\partial}{\partial \iota_j} \log p(Y_T|\lambda) = \left(\frac{p(s_0 = j|Y_T, \lambda)}{\Phi(\rho_j)} + \frac{p(s_0 = K|Y_T, \lambda)}{1 - \sum_{i=1}^{K-1} \Phi(\rho_i)} \right) \cdot \phi(\rho_j) \text{ for } \rho_j \in \iota \qquad \text{(EQ 40)}$$

The third case is the trickiest. Here, we initialize the filter using the unconditional state probabilities, so

$$p(s_0|\rho) = p(s|\rho) \quad \text{(EQ 32)}$$

Note that we have valid formulas for these probabilities only in the constant transition probability case.[9] When we allow for time variation in the transition probabilities, we approximate by replacing the constant transition probability matrix P with the mean transition probabilities

$$\bar{P} = \left(\sum_{t=1}^{T} P_t \right) / T \quad \text{(EQ 33)}$$

where the *ij*th element of the square matrix P_t is

$$P_{ijt} = \Phi(z_t \cdot \gamma_{ij}) \quad \text{(EQ 34)}$$

Hamilton (1994) shows that these unconditional probabilities $p(s|\rho)$ may be found as the elements of the eigenvector of P with the eigenvalue 1. This requires us to find the derivative of a particular eigenvector with respect to elements in the matrix $\bar{P}$, which is difficult and might be easier to approximate using numerical methods.

In the two-state case, however, simpler solutions are obtainable. In particular,

$$p(1|\rho) = \frac{1-\bar{p}_{22}}{2-\bar{p}_{11}-\bar{p}_{22}}, \; p(2|\rho) = \frac{1-\bar{p}_{11}}{2-\bar{p}_{11}-\bar{p}_{22}} \quad \text{(EQ 35)}$$

Note that because these two expressions sum to one, it follows that

$$\frac{d}{d\bar{p}_{11}} p(1|\rho) = -\frac{d}{d\bar{p}_{11}} p(2|\rho), \; \frac{d}{d\bar{p}_{22}} p(1|\rho) = -\frac{d}{d\bar{p}_{22}} p(2|\rho)$$

It is straightforward to show that

$$\frac{\partial}{\partial \gamma_{11}} p(1|\rho) = \frac{\sum_{t=1}^{T} \phi(z_t \cdot \gamma_{11}) \cdot z_t}{T \cdot (2-\bar{p}_{11}-\bar{p}_{22})}, \quad \frac{\partial}{\partial \gamma_{22}} p(2|\rho) = \frac{\sum_{t=1}^{T} \phi(z_t \cdot \gamma_{22}) \cdot z_t}{T \cdot (2-\bar{p}_{11}-\bar{p}_{22})} \quad \text{(EQ 36)}$$

9. See Hamilton (1994), Chap 22.2. He refers to these “unconditional” probabilities as the *ergodic* probabilities.

the full-sample smoother. The term $p(s_t|s_{t-1}, \rho)$ is defined in (EQ 3) and its derivative is simply

$$\frac{\partial}{\partial \rho_j} p(s_t|s_{t-1}, \rho) = \phi(z_t \cdot \gamma_{s_t s_{t-1}}) \cdot z_t \text{ if } \rho_j \in \gamma_{s_t s_{t-1}} \text{ and 0 otherwise.} \quad \text{(EQ 28)}$$

In applying this formula, it is important to keep in mind that each ρ_j enters in γ_{ik} for at least two different values of *i* given any value of *k*. This simply reflects the fact that

$$\sum_{i=1}^{K} p(s_t = i|s_{t-1} = k) = 1 \quad \text{(EQ 29)}$$

Therefore, if changes in ρ_j affect $p(s_t = i|s_{t-1} = k)$, they must also affect some $p(s_t = l|s_{t-1} = k)$ for some $l \in [1, K]$ and $l \neq i$ in order for the sum of probabilities across all *K* states to be equal to 1. Applying (EQ 28) therefore requires us to specify how we ensure that our parameterization obeys (EQ 29). We will assume that there is some other parameter $\rho_{l \neq j}$ that will be adjusted to satisfy (EQ 29). We will return to this point below.

The derivative of $\log p(s_0|\rho)$ is not straightforward, since it will depend on how we choose to initialize the filter. While there are a variety of ways to do so, we need worry about only three cases.

In the first case, we simply assign a set of initial state probabilities, so

$$p(s_0|\rho) = p(s_0)$$

$$\therefore \frac{\partial}{\partial \rho_k} \log p(s_0|\rho) = 0. \quad \text{(EQ 30)}$$

In the second case, we treat the initial state probabilities as part of the optimization problem, so we add an extra *K-1* parameters (call them ι) to ρ such that

$$p(s_0|\rho) = p(s_0|\iota) = \Phi(\iota)$$

$$\therefore \frac{\partial}{\partial \rho_k} \log p(s_0|\rho) = \phi(\iota)/\Phi(\iota) \text{ if } \rho_k \in \iota \text{ and 0 otherwise.} \quad \text{(EQ 31)}$$

Again, note that in this case we will need to be careful to respect (EQ 29).

$$\frac{\partial}{\partial \beta_{ij}} p(y_t | s_t = k, \theta) = 0 \qquad \text{(EQ 22)}$$

$$\frac{\partial}{\partial \sigma_i} p(y_t | s_t = k, \theta) = 0 \qquad \text{(EQ 23)}$$

Finally, $p(s_t | Y_T, \lambda)$ is simply the smoothed probability of having observed s_t.

If we now substitute some of these back into (EQ 14), we obtain

$$\frac{\partial}{\partial \beta_{ij}} \log p(Y_T | \lambda) = \sum_{t=1}^{T} \left\{ \frac{\frac{\partial}{\partial \beta_{ij}} p(y_t | s_t = i, \theta)}{p(y_t | s_t = i, \theta)} \cdot p(s_t = i | Y_T, \lambda) \right\} \qquad \text{(EQ 24)}$$

$$\therefore \frac{\partial}{\partial \beta_{ij}} \log p(Y_T | \lambda) = \sum_{t=1}^{T} \left(\frac{y_t - x_t \cdot \beta_i}{\sigma_i} \right) \cdot \left(\frac{x_{jt}}{\sigma_i} \right) \cdot p(s_t = i | Y_T, \lambda) \qquad \text{(EQ 25)}$$

$$\frac{\partial}{\partial \sigma_i} \log p(Y_T | \lambda) = \sum_{t=1}^{T} \left\{ \frac{\frac{\partial}{\partial \sigma_i} p(y_t | s_t = i, \theta)}{p(y_t | s_t = i, \theta)} \cdot p(s_t = i | Y_T, \lambda) \right\} \qquad \text{(EQ 26)}$$

$$\therefore \frac{\partial}{\partial \sigma_i} \log p(Y_T | \lambda) = \sum_{t=1}^{T} \left(\frac{(y_t - x_t \cdot \beta_i)^2}{\sigma_i^2} - 1 \right) \cdot \frac{p(s_t = i | Y_T, \lambda)}{\sigma_i} \qquad \text{(EQ 27)}$$

We conclude this section with some observations on computational efficiency. First, since all of these derivatives require both the regime-dependent residuals $(y_t - x_t \cdot \beta_i)$ and the smoother probabilities, it makes sense to calculate these common elements before calculating the individual derivatives. Second, (EQ 25) implies that the derivative of two slope coefficients in the same state differ only by a factor of x_{it} / x_{jt}, suggesting a shortcut that could be used after calculating the derivative with respect to β_{i1}.

4.5 $\frac{\partial}{\partial \rho_j} p(S_T | \rho)$

From (EQ 16), we see that we have four terms to contend with: $p(s_t | Y_T, \lambda)$, $\frac{\partial}{\partial \rho_j} p(s_t | s_{t-1}, \rho)$, $\frac{\partial}{\partial \rho_j} \log p(s_0 | \rho)$, and $p(s_0 | Y_T, \lambda)$. Of these, $p(s_t | Y_T, \lambda)$ is calculated via

This is further simplified by taking the derivative of the log, which gives

$$\frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \left(\sum_{s_0} p(s_0|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_0|\rho) \right) + \sum_{t=1}^{T} \sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} \left(\frac{p(s_{t-1}|\lambda, Y_t)\, p(s_t|Y_T, \lambda)}{p(s_t|\lambda, Y_t)} \right) \frac{\partial}{\partial \rho_j} p(s_t|s_{t-1}, \rho) \quad \text{(EQ 16)}$$

4.4 $\frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda)$

We already know how to calculate several of the elements in (EQ 14). Once we condition on the state s_t (and, implicitly, the independent variables x_t), y_t has a normal distribution, so

$$p(y_t|s_t, \theta) = \phi\left(\frac{y_t - x_t \cdot \beta_{s_t}}{\sigma_{s_t}}\right) \cdot \sigma_{s_t}^{-1} \quad \text{(EQ 17)}$$

Since

$$\phi(x) \equiv \frac{e^{-x^2/2}}{\sqrt{2\pi}} \quad \text{(EQ 18)}$$

it follows that

$$\frac{d}{dx}\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}} \cdot (-x) = -x \cdot \phi(x) \quad \text{(EQ 19)}$$

This means that

$$\frac{\partial}{\partial \beta_{ij}} p(y_t|s_t = i, \theta) = \phi\left(\frac{y_t - x_t \cdot \beta_i}{\sigma_i}\right) \cdot \left(\frac{y_t - x_t \cdot \beta_i}{\sigma_i} \right) \cdot \frac{x_{it}}{\sigma_i^2} \quad \text{(EQ 20)}$$

and

$$\frac{\partial}{\partial \sigma_i} p(y_t|s_t = i, \theta) = \phi\left(\frac{y_t - x_t \cdot \beta_i}{\sigma_i}\right) \cdot \sigma_i^{-2} \cdot \left(\frac{(y_t - x_t \cdot \beta_i)^2}{\sigma_i^2} - 1 \right) \quad \text{(EQ 21)}$$

Furthermore, it should be clear that when $k \neq i$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \left(\sum_{s_0} p(s_0|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_0|\rho) \right) + \sum_{t=1}^{T} \sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) \quad \text{(EQ 15)}$$

Since $\sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) =$

$$\sum_{s_1=1}^{K} \cdots \sum_{s_T=1}^{K} p(s_1, \ldots, s_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) =$$

$$\sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} \sum_{s_0=1}^{K} \cdots \sum_{s_{t-2}=1}^{K} \sum_{s_{t+1}=1}^{K} \cdots \sum_{s_T=1}^{K} p(s_1, \ldots, s_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho)$$

and this last expression can be rewritten as

$$\sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) \cdot \left(\sum_{s_0=1}^{K} \cdots \sum_{s_{t-2}=1}^{K} \sum_{s_{t+1}=1}^{K} \cdots \sum_{s_T=1}^{K} p(s_1, \ldots, s_T|Y_T, \lambda) \right) =$$

$$\sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) \cdot p(s_t, s_{t-1}|Y_T, \lambda),$$

(EQ 15) can be rewritten as

$$\frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{s_0} p(s_0|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_0|\rho) + \sum_{t=1}^{T} \sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} p(s_t s_{t-1}|Y_T, \lambda) \frac{\partial}{\partial \rho_j} (\log p(s_t|s_{t-1}, \rho))$$

Then $p(s_t, s_{t-1}|Y_T, \lambda)$ can be broken down to give[8]

$$\sum_{t=1}^{T} \sum_{s_t=1}^{K} \sum_{s_{t-1}=1}^{K} \left(\frac{p(s_t|s_{t-1}, \rho)\, p(s_{t-1}|\lambda, Y_t)\, p(s_t|Y_T, \lambda)}{p(s_t|\lambda, Y_t)} \right) \frac{\partial}{\partial \rho_j} (\log p(s_t|s_{t-1}, \rho))$$

8. For a proof, see Hamilton (1994), p. 701.

The derivation of the analytical derivatives now requires only two major steps: collapsing the $\sum_{S_T}$ expression and then simplifying the remaining terms in (EQ 10) and (EQ 11).

4.3 Collapsing the Summation

To understand how we collapse the summation, consider first the simpler problem of collapsing the summation in the following expression:

$$\sum_{S_T}\left\{\sum_{t=1}^{T}\frac{\frac{\partial}{\partial\theta_j}p(y_t|s_t,\theta)}{p(y_t|s_t,\theta)}\right\} = \sum_{s_1=1}^{K}\cdots\sum_{s_T=1}^{K}\left\{\sum_{t=1}^{T}\frac{\frac{\partial}{\partial\theta_j}p(y_t|s_t,\theta)}{p(y_t|s_t,\theta)}\right\} \qquad \text{(EQ 12)}$$

(EQ 12) shows the expansion for the general case of a *K* state switching model. Note that the *T* independent summations of *K* states each will produce all K^T possible permutations of the vector of realizations S_T. However, the right-hand term in parentheses is a function of only s_t, which implies that of the K^T terms in the summation over all possible S_T, we have at most only *K* distinct terms. In other words, taking together all the *T+1* summations in (EQ 12), we have $K^T \cdot T$ terms, of which at most $K \cdot T$ are distinct. This means that (EQ 12) can be rewritten as

$$K^{T-1}\cdot\sum_{t=1}^{T}\left\{\sum_{s_t=1}^{K}\frac{\frac{\partial}{\partial\theta_j}p(y_t|s_t,\theta)}{p(y_t|s_t,\theta)}\right\} \qquad \text{(EQ 13)}$$

(EQ 10) differs from (EQ 12) only in that each term in the summation over S_T is weighted by the "smoothed" probability of its occurrence. However, since these probabilities are themselves functions of s_t only rather than all of S_T, we can rewrite (EQ 10) as[7]

$$\frac{\partial}{\partial\theta_j}\log p(Y_T|\lambda) = \sum_{t=1}^{T}\left\{\sum_{s_t=1}^{K}\left\{\frac{\frac{\partial}{\partial\theta_j}p(y_t|s_t,\theta)}{p(y_t|s_t,\theta)}\cdot p(s_t|Y_T,\lambda)\right\}\right\} \qquad \text{(EQ 14)}$$

For $\lambda_j \in \rho$, we will first rewrite the order of summation

7. This can be shown on induction on *t*.

it follows that

$$\frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda) = \sum_{S_T} \left(\prod_{t=1}^{T} p(s_t|Y_T, \lambda) \right) \cdot \left\{ \frac{\partial}{\partial \theta_j} \sum_{t=1}^{T} \log p(y_t|s_t, \theta) \right\}$$

$$\therefore \frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda) = \sum_{S_T} \left(\prod_{t=1}^{T} p(s_t|Y_T, \lambda) \right) \cdot \left\{ \sum_{t=1}^{T} \frac{\frac{\partial}{\partial \theta_j} p(y_t|s_t, \theta)}{p(y_t|s_t, \theta)} \right\} \qquad \text{(EQ 10)}$$

Similarly, (EQ 5) and (EQ 6) imply that for $\lambda_j \in \rho$, we obtain

$$\frac{\partial}{\partial \lambda_j} \log p(Y_T|\lambda) = \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \frac{1}{p(Y_T|\lambda)} \cdot \frac{\partial}{\partial \rho_j} \sum_{S_T} p(Y_T|S_T, \theta) \cdot p(S_T|\rho)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \frac{1}{p(Y_T|\lambda)} \cdot \sum_{S_T} p(Y_T|S_T, \theta) \cdot \frac{\partial}{\partial \rho_j} p(S_T|\rho)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{S_T} \frac{p(Y_T, S_T|\lambda)}{p(Y_T|\lambda) \cdot p(S_T|\rho)} \cdot \frac{\partial}{\partial \rho_j} p(S_T|\rho) = \sum_{S_T} \frac{p(S_T|Y_T, \lambda)}{p(S_T|\rho)} \cdot \frac{\partial}{\partial \rho_j} p(S_T|\rho)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(S_T|\rho)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log \left(p(s_0|\rho) \cdot \prod_{t=1}^{T} p(s_t|s_{t-1}, \rho) \right)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \left(\log p(s_0|\rho) + \sum_{t=1}^{T} \log p(s_t|s_{t-1}, \rho) \right)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \left(\sum_{S_T} p(S_T|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_0|\rho) \right) +$$

$$\sum_{S_T} p(S_T|Y_T, \lambda) \cdot \left(\sum_{t=1}^{T} \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) \right)$$

$$\therefore \frac{\partial}{\partial \rho_j} \log p(Y_T|\lambda) = \left(\sum_{s_0} p(s_0|Y_T, \lambda) \cdot \frac{\partial}{\partial \rho_j} \log p(s_0|\rho) \right) +$$

$$\sum_{S_T} p(S_T|Y_T, \lambda) \cdot \left(\sum_{t=1}^{T} \frac{\partial}{\partial \rho_j} \log p(s_t|s_{t-1}, \rho) \right) \qquad \text{(EQ 11)}$$

4.2 Basic Results

We are trying to find $\frac{\partial}{\partial \lambda_j} \log p(Y_T|\lambda)$. Obviously,

$$\frac{\partial}{\partial \lambda_j} \log p(Y_T|\lambda) = \frac{1}{p(Y_T|\lambda)} \cdot \frac{\partial}{\partial \lambda_j} p(Y_T|\lambda) \quad \text{(EQ 5)}$$

Note that we can rewrite the likelihood function as

$$p(Y_T|\lambda) = \sum_{S_T} p(Y_T|S_T, \lambda) \cdot p(S_T|\lambda) = \sum_{S_T} p(Y_T|S_T, \theta) \cdot p(S_T|\rho) \quad \text{(EQ 6)}$$

where the first equality follows from Bayes Theorem and the second from the properties of the switching model.

This means for $\lambda_j \in \theta$,

$$\frac{\partial}{\partial \lambda_j} p(Y_T|\lambda) = \frac{\partial}{\partial \theta_j} p(Y_T|\lambda) = \frac{\partial}{\partial \theta_j} \sum_{S_T} p(Y_T|S_T, \theta) \cdot p(S_T|\rho) = \sum_{S_T} p(S_T|\rho) \cdot \frac{\partial}{\partial \theta_j} p(Y_T|S_T, \theta) \quad \text{(EQ 7)}$$

Using (EQ 5) then gives

$$\frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda) = \frac{1}{p(Y_T|\lambda)} \cdot \sum_{S_T} p(S_T|\rho) \cdot \frac{\partial}{\partial \theta_j} p(Y_T|S_T, \theta)$$

$$\therefore \frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda) = \sum_{S_T} \left(\frac{p(Y_T|S_T, \theta) \cdot p(S_T|\rho)}{p(Y_T|\lambda)} \right) \cdot \left(\frac{\partial}{\partial \theta_j} \log p(Y_T|S_T, \theta) \right)$$

$$\therefore \frac{\partial}{\partial \theta_j} \log p(Y_T|\lambda) = \sum_{S_T} p(S_T|Y_T, \lambda) \cdot \left(\frac{\partial}{\partial \theta_j} \log p(Y_T|S_T, \theta) \right) \quad \text{(EQ 8)}$$

Since

$$p(Y_T|S_T, \theta) = \prod_{t=1}^{T} p(y_t|s_t, \theta),$$

$$p(S_T|Y_T, \lambda) = \prod_{t=1}^{T} p(s_t|Y_T, \lambda) \quad \text{(EQ 9)}$$

4.0 Mathematical Appendix: Derivation of Analytical Gradients

The general regime-switching model that we consider describes the generation of a single variable y_t by *K* distinct states. In any given state *i*, y_t is generated by a linear regression model

$$y_t = x_t \cdot \beta_i + \varepsilon_{it},\ \varepsilon_{it} \sim \text{i.i.d. } N(0, \sigma_i) \qquad \text{(EQ 3)}$$

where x_t is a vector that is exogenous with respect to ε_{it}. The states are assumed to follow a first-order Markov process, with transition probabilities that may vary over time according to the formula[6]

$$P(s_t = i \,|\, s_{t-1} = j) = \Phi(z_t \cdot \gamma_{ij}) \qquad \text{(EQ 4)}$$

where Φ is the standard normal cumulative distribution.

4.1 Notation

Let ϕ and Φ denote the standard normal probability density function and cumulative distribution function.

Let Y_T, X_T, Z_T, S_T respectively denote a matrix of values from time *1* to *T* of y_t, x_t, z_t, s_t.

Let the sum of $f(S_T)$ over all possible realizations of S_T (which in general has K^T permutations) define the expression $\sum_{S_T} f(S_T)$.

Let the parameters of the model be gathered into a vector $\lambda' = \left[\theta\ \rho\right]'$, where ρ captures all the parameters entering into the state-transition probabilities (EQ 3), and θ is all the other model parameters. We denote the *j*th element of λ by λ_j.

The likelihood function of this model is $p(Y_T | \lambda, X_T, Z_T)$, which for simplicity we will write as $p(Y_T | \lambda)$.

6. We assume that z_t is also exogenous with respect to ε_{it}.

The analytical derivatives make one call to the smoother, which in turn calculates the likelihood function and then performs an additional loop over the sample size. Hence, the smoother requires approximately 2*N calculations. Calculating the gradient then requires a loop over the number of states (S) for a set number of calculations (L). In total, therefore, this requires approximately 2*N+S*L calculations.

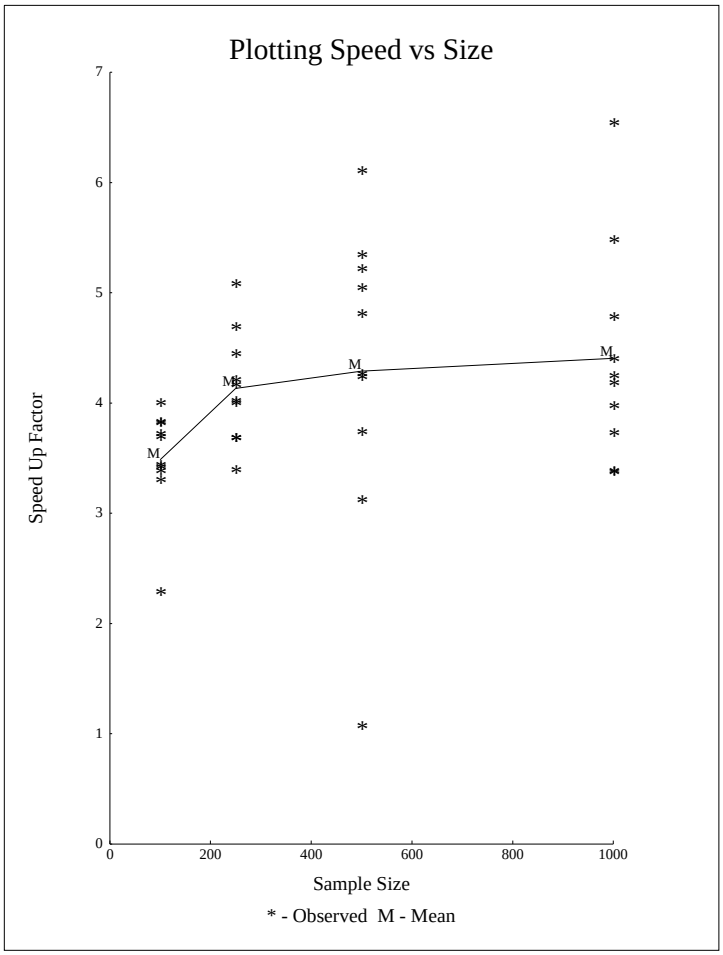
With the above analysis, we can predict that the analytical gradient procedure will be faster as either the sample size or the number of parameters increases. In the unlikely case that the number of states increases with the number of parameters held constant, there may be a relative improvement of the numerical technique. More likely, an increase in the number of states will cause a substantial increase in the number of parameters.

These conclusions will of course depend on the relative costs (time required) of different operations. For interpreted languages (like Gauss), loops are particularly inefficient and operations should be vectorized wherever possible. If our procedure were rewritten in a compiled language, the relative efficiency of numerical and analytical gradients for a particular model might change. However, we would still expect that the relative efficiency of the analytical gradients would increase with the number of parameters.

A minimum of 1.069 signifies that for one draw the analytical gradients were only negligibly faster than the numerical.

In general, we see that analytical derivatives can improve calculation efficiency by roughly a factor of four. The relative speed of the analytical derivatives increases with an increase in sample size, but the improvement seems to slow as samples become large. Results are also somewhat variable. For example, in one experiment with 500 observations, analytical gradients produced almost no time savings relative to the numerical gradients. However, of the remaining nine cases examined for this sample size, the speed-up factor was never less than three.

Figure 1



An analysis of the code illustrates why the analytical derivatives are typically faster than the numerically calculated derivatives. The numerically calculated derivatives make P (the number of parameters) calls to the likelihood function. Most of the time in the likelihood function is in turn spent in a loop over the sample size (N). One therefore could approximate the amount of calculations in the numerical derivatives as P*N.

First, we note that for each parameter, the median percentage difference in the estimates using the two kinds of gradients was less than 10^{-7}. There were a few larger differences, however. Upon further examination, we found that most of the large discrepancies came from a single sample. Removing this data sample (#71) from our experiment then produced the results shown in Table 3. Mean, minimum and maximum differences between the two sets of parameter estimates are now much reduced for most parameters. Furthermore, we found that the problem with the 71st sample appeared to be one of two local maximums. If either set of parameters is supplied as the starting values, both analytical and numerical gradients will converge to that value.

Based on the above results, we can conclude that the analytical and the numerical gradients will generally converge to the same estimate. However, the sensitivity of maximum likelihood estimation to intermediate calculations when there are multiple local maximums may occasionally cause differences. This does not imply that one method is superior to another. It simply means that researchers should remember to check for multiple local maximums.

TABLE 3. Results After Removing the 71st Trial

Parameter	MEAN	MEDIAN	MAX.	MIN.
β_{11}	6.1340217e-07	0.0000000	7.0871000e-05	-3.0400000e-05
β_{12}	-8.5868450e-06	1.6933548e-08	0.00012254550	-0.00054090740
β_{13}	5.3470534e-05	0.0000000	0.0092465532	-0.00099279136
β_{21}	2.6145186e-06	0.0000000	0.00035430604	-0.00015096286
β_{22}	-1.3683861e-07	5.2138270e-08	0.00052705071	-0.00075399754
β_{23}	-0.0096435665	3.2498027e-08	0.024998039	-0.89744424
$\gamma_{11,\,1}$	-0.0014503071	0.0000000	8.6297094e-05	-0.058690961
$\gamma_{11,\,1}$	-3.5319578e-06	4.2514594e-08	8.6125427e-05	-0.00034058177
σ	4.0692540e-09	0.0000000	1.1172225e-05	-7.7739852e-06

3.0 Speed

To compare the computational efficiency of the two gradient procedures, we again estimated regime-switching models on simulated data, using both procedures on each data set. This was done 10 times for four different sample sizes. Results are reported in Table 4 and in Figure 1.

TABLE 4. SPEED UP FACTOR:
(Total Elapsed Time with Numerical Gradients) / (Total Elapsed Time with Analytical Gradients)

# Obs.	MEAN	MAX.	MIN.	Standard Deviation
100	3.4840708	3.9978395	2.2788343	0.48204014
250	4.1314662	5.0734526	3.3856693	0.50488882
500	4.2898862	6.0966392	1.0698553	1.4188306
1000	4.4052682	6.5387418	3.3724443	0.98204026

TABLE 1. Accuracy of Analytical versus Numerical Gradients - Example *(continued)*

Parameter[a]	Computed Gradient	Forward Gradient	Central Gradient	Richardson Gradient
β_{22}	0.05427607	0.05427607	0.05427597	0.05427675
β_{23}	0.00416917	0.00416917	0.00416920	0.00416880
β_{24}	-0.00612684	-0.00612684	-0.00612680	-0.00612742
$\gamma_{11,\,1}$	0.04583327	0.04583326	0.04583328	0.04583299
$\gamma_{11,\,1}$	0.04583327	0.04583327	0.04583331	0.04583299
$\gamma_{22,\,1}$	-0.01941340	-0.01941340	-0.01941342	-0.01941372
$\gamma_{22,\,2}$	-0.01941340	-0.01941340	-0.01941336	-0.01941372
σ_1	-0.32985335	-0.32985335	-0.32984718	-0.32985359
σ_2	-0.14088036	-0.14088036	-0.14087911	-0.14088066

a. β_{1j} and β_{2j} are the coefficients of the linear regressions for states 1 and 2. $\gamma_{kk,\,i}$ are the coefficients for the transitions probability for remaining in state k. σ_k is the standard deviations of the error term associated with state k.

Based on these results, we should expect analytical and numerical gradients to produce the same results when used for parameter estimation. We checked this using a small Monte Carlo experiment. A two-state switching regression was estimated using both methods[5] for one hundred samples of a hundred observations each. Table 2 reports the difference between the two sets of parameter estimates.

TABLE 2. Percentage Differences of Maximum Likelihood Parameter Estimates

Parameter[a]	MEAN	MEDIAN	MAX	MIN
β_{11}	0.0014182477	0.0000000	0.13184060	-3.0400000e-05
β_{12}	-0.0015846355	1.0105265e-08	0.00012254550	-0.14658111
β_{13}	-0.0089365709	0.0000000	0.0092465532	-0.83602038
β_{21}	0.0018295629	0.0000000	0.16990881	-0.00015096286
β_{22}	-0.0033434991	4.1351604e-08	0.00052705071	-0.31093283
β_{23}	-0.014223531	2.9252218e-08	0.024998039	-0.89744424
$\gamma_{11,\,1}$	-0.0039566721	0.0000000	8.6297094e-05	-0.23454226
$\gamma_{11,\,1}$	-0.0068314097	3.3581302e-08	8.6125427e-05	-0.63499616
σ	-0.00087577934	0.0000000	1.1172225e-05	-0.081447853

a. Notation as per Table 1 except here the standard deviation of the error term σ is constrained to be the same across states. The data was generated and then estimated as being $y_t = \beta_1 + \beta_2 X_{2,\,t} + \beta_3 X_{3,\,t} + 1.4\varepsilon_t$, where $X_{i,\,t}$ and ε_t are i.i.d N(0,1). The probability of remaining in regime i is $\Phi(z_{ii})$ The two regimes' coefficient vectors $[\beta_1, \beta_2, \beta_3, z_{ii}]$ are $[0.6, 0.7, 0.5, 1.8]$ and $[0.2, -0.5, 0.3, 0.6]$.

5. Maxlik's default, the central difference method, did the numerical calculation of the gradients.

(1994). Fourth, Durland and McCurdy (1994) show that such models can usefully approximate a duration-dependent semi-Markov process.

Regime-switching models have become popular for the modelling of business cycles, as originally proposed by Hamilton (1989).[4] It may therefore seem surprising that there has been no exposition of how to calculate the derivatives analytically. There are two reasons for this. First, analytical derivatives are available for a more restrictive class of models (Hamilton 1992). Second, and perhaps more importantly, analytical derivatives were thought to be of little practical use in estimation. This is because analytical calculation of the score requires (as we will see) calculation of the smoothed probabilities of being in each possible state *i* at each time *t*. These smoothed probabilities were in turn much more costly to compute than the likelihood function, which made the alternative of numerical derivatives attractive. However, Kim (1994) presented a new algorithm for calculation of these smoothed probabilities that can reduce calculation times by as much as several orders of magnitude. This in turn raises the possibility that analytical gradients may now be more efficient than numerical methods for a broad class of regime-switching models.

This paper is organized as follows. Due to its length, the formal derivation of the analytical derivatives of our model is presented in a mathematical appendix. A second appendix lists three Gauss procedures that calculate the smoothed probabilities, the likelihood function and the score vector (respectively) for the case where K=2. The remainder of this paper compares the efficiency of numerical and analytical gradients in maximizing the likelihood function for a variety of models. The first section below considers the accuracy of the analytical and numerical gradients and the consistency of the parameter estimates. The section thereafter compares the speed of calculation.

2.0 Accuracy

The accuracy of the gradient calculations was tested on a sample of 100 observations for arbitrary parameter vectors. We found that the analytical gradients were identical to those provided by the forward gradient technique and differed only very slightly from the results of the two other numerical methods. Sample results are provided in Table 1.

TABLE 1. Accuracy of Analytical versus Numerical Gradients - Example

Parameter[a]	Computed Gradient	Forward Gradient	Central Gradient	Richardson Gradient
β_{11}	0.06495304	0.06495304	0.06495437	0.06495362
β_{12}	0.03838056	0.03838056	0.03838199	0.03838091
β_{13}	0.02906685	0.02906685	0.02906791	0.02906681
β_{14}	-0.04750326	-0.04750326	-0.04750253	-0.04750267
β_{21}	0.01263473	0.01263473	0.01263476	0.01263431

4. Note that Hamilton's (1989) two-state model with q autoregressive lags may be written in this framework as a model with $2 \cdot q$ states.

1.0 Introduction

This paper derives analytical derivatives for a broad class of regime-switching models with Markovian state-transition probabilities. Estimation of these models is usually done by maximum likelihood methods, which require some way of calculating the derivatives of the likelihood function with respect to the parameter vector.[1] (This vector is also referred to as the *score* or the *gradient* vector.) This is usually done using numerical techniques that approximate the derivative by the change in the likelihood function for small changes in the parameter vector. This is not especially efficient, however, as such techniques typically require *N+1* evaluations of the likelihood function to calculate the *N* elements of the score, and $N^2 + 1$ to calculate the Hessian (the matrix of second derivatives). By using analytical gradients, we show that the number of calculations required to evaluate either of these objects can be greatly reduced. This in turn considerably speeds up maximum-likelihood estimation of such models with no loss in accuracy.

The general regime-switching model that we consider describes the generation of a single variable y_t by *K* distinct states. In any given state *i*, y_t is generated by a linear regression model

$$y_t = x_t \cdot \beta_i + \varepsilon_{it},\ \varepsilon_{it} \sim \text{i.i.d. } N(0, \sigma_i) \qquad \text{(EQ 1)}$$

where x_t is a vector that is exogenous with respect to ε_{it}. The states are assumed to follow a first-order Markov process, with transition probabilities that may vary over time according to the formula[2]

$$P(s_t = i \,|\, s_{t-1} = j) = \Phi(z_t \cdot \gamma_{ij}) \qquad \text{(EQ 2)}$$

where $\Phi(\,)$ is the standard normal cumulative distribution function.

This class of models encompasses many useful special cases. First, it should be noted that any higher *p*th order Markov process with *K* states can be written as a first-order Markov process with $p \cdot K$ states, so our assumption that the process is first-order is not restrictive. Second, the case where z_t is a scalar constant gives us the Markov switching regressions originally proposed by Goldfeld and Quandt (1973) and discussed in Hamilton (1994).[3] Third, the case where x_t is a scalar constant is explored by Diebold, Lee and Weinbach

1. Alternatives to maximum likelihood estimation include the EM algorithm and simulated annealing. However, simulated annealing is also inefficient, and the EM algorithm has a lower rate of convergence near the optimum than does most popular gradient-based maximization methods (such as the Newton algorithm.) The score is also useful for the calculation of standard errors, and for diagnostic tests (See White (1994).)

2. We assume that z_t is also exogenous with respect to ε_{it}.

3. Hamilton (1994) also discusses the case where y_t may be a vector rather than a scalar.

Table of Contents

Abstract

This paper derives analytical gradients for a broad class of regime-switching models with Markovian state-transition probabilities. Such models are usually estimated by maximum likelihood methods, which require the derivatives of the likelihood function with respect to the parameter vector. These gradients are usually calculated by means of numerical techniques. The paper shows that analytical gradients considerably speed up maximum-likelihood estimation with no loss in accuracy. A sample program listing is included.

Résumé

Dans cette étude, les auteurs dérivent des gradients analytiques pour toute une catégorie de modèles à changement de régime comportant des probabilités de transition à la Markov. Ces modèles sont généralement estimés à l'aide de méthodes du maximum de vraisemblance, qui nécessitent que la fonction de vraisemblance soit dérivée par rapport au vecteur des paramètres du modèle. Les gradients sont habituellement calculés à l'aide de techniques numériques. Les auteurs montrent que l'utilisation de gradients analytiques accélère considérablement les estimations effectuées à l'aide des méthodes du maximum de vraisemblance, sans toutefois nuire à leur précision. Un imprimé du programme informatique est fourni à la fin de l'étude.

ISSN 1192-5434
ISBN 0-662-23685-8

Printed in Canada on recycled paper

August 1995

Analytical Derivatives for Markov Switching Models

Jeff Gable
Queen's University, Kingston, Ontario, Canada[1]

Simon van Norden
E-mail: svannorden@bank-banque-canada.ca
International Department, Bank of Canada
Ottawa, Ontario, Canada K1A 0G9

Robert Vigfusson
E-mail: rvigfusson@bank-banque-canada.ca
International Department, Bank of Canada
Ottawa, Ontario, Canada K1A 0G9

1. This paper was written while this author was with the International Department of the Bank of Canada.

Working Paper 95-7 / Document de travail 95-7

Analytical Derivatives for Markov Switching Models

by
Jeff Gable, Simon van Norden
and Robert Vigfusson

Bank of Canada  Banque du Canada