```
// T51mixExpNor1.sce
// MIXTURE ESTIMATION (descriptive normal)
// - static normal componens
// - scalar y and multinomial continuous v
//   |y| =  |a1 a2 a3||v1| + e
//                   |v2|
//                   |v3|
// Experiments
// - change simulated expectations thS
// - change initial expectations thE (through the statistics S, ka)
// ---------------------------------------------------------------------
exec("ScIntro.sce",-1),
getd(), mode(0)

nd=100;                                                          // 1
// SIMULATION                                                    // 2
thS=list();                                                      // 3
thS(1)=[1 6 10];                       // simulated regr. coefficants  // 4
thS(2)=[8 1 5];                                                  // 5
thS(3)=[6 9 1];                                                  // 6
sd=.1;                                 // common and known noise std.  // 7
for t=1:nd                                                       // 8
  jS=ceil(3*randu());                  // pointer value generation     // 9
```

```
  cS(t)=jS;                                                    // 10
  v(t,:)=randn(1,3);                    // explan. variables    // 11
  y(t)=thS(jS)*v(t,:)'+sd*randn();      // output value generation // 12
end                                                            // 13
nc=length(thS);                                                // 14
                                                               // 15
// INITIALIZATION                                              // 16
V=list(); thE=list();                                          // 17
ka=[1 1 1];                             // initial counter     // 18
for j=1:nc                                                     // 19
  thE(j)=thS(j)+5*rand(1,3);            // initial parameters  // 20
  V(j)=eye(4,4);                        // initial inf. matrix // 21
  V(j)(1,2:4)=thE(j)';                                         // 22
  V(j)(2:4,1)=thE(j);                                          // 23
  thE(j)=v2thN(V(j)/ka(j),1);           // initial point estimates // 24
end                                                            // 25
                                                               // 26
// TIME LOOP                                                   // 27
for t=1:nd                                                     // 28
  // estimation                                                // 29
  for j=1:nc                                                   // 30
    [nill,qp(j)]=GaussN(y(t),thE(j)'*v(t,:)',sd); // proximity // 31
  end                                                          // 32
```

```
  q=exp(qp-max(qp));                    // loq q -> w                  // 33
  w=q/sum(q);                           // weights                     // 34
  wt(:,t)=w;                                                        // 35
  for j=1:nc                                                        // 36
    Ps=[y(t) v(t,:)]';                  // extended reg. vector        // 37
    V(j)=V(j)+w(j)*Ps*Ps';              // update of statistics V      // 38
    ka(j)=ka(j)+w(j);                   //   and counter               // 39
    thE(j)=v2thN(V(j)/ka(j),1);         // point estimates             // 40
    c(j).th(:,t)=thE(j);                // remember for plot           // 41
  end                                                               // 42
  // zero-step prediction                                           // 43
  ypp=0;                                                            // 44
  for j=1:nc                                                        // 45
    select 1        // type of prediction: 1-point, 2-generated;    // 46
    case 1, ypp=ypp+w(j)*thE(j)'*v(t,:)';                           // 47
    case 2, ypp=ypp+w(j)*(thE(j)'*v(t,:)'+sd*randn());              // 48
    end                                                             // 49
  end                                                               // 50
  yp(t)=ypp;                            // prediction                  // 51
end                                                                 // 52
                                                                    // 53
// RESULTS                                                          // 54
set(scf(1),'position',[600 10 1200 400])                            // 55
```

```
for j=1:nc                                                          // 56
  subplot(1,3,j)                                                    // 57
  plot(c(j).th')                                                    // 58
  title('Evolution of estimated parametrs: comp. '+string(j))       // 59
end                                                                 // 60
                                                                    // 61
disp 'Simulated parameter values'                                   // 62
disp(thS(:))                                                        // 63
disp 'Final estimated parameters'                                   // 64
disp(thE(:))                                                        // 65
                                                                    // 66
set(scf(2),'position',[900 500 600 400])  // plot data and prediction // 67
plot(1:nd,y,'ob',1:nd,yp,'xr','markersize',8)                       // 68
title 'Data (b) and prediction (r)'                                 // 69
                                                                    // 70
[nill,cp]=max(wt,'r');                    // accuracy of classification // 71
disp 'Accuracy of classification'                                   // 72
ACC=acc(cS,cp)                                                      // 73
                                                                    // 74
disp 'Relative prediction error'          // accuracy of prediction  // 75
RPE=rpe(y(:),yp(:))                                                 // 76
```

Description of the program

- Rows 3–7 define parameters for simulation.
- Rows 9–12 perform simulation with switching components. Simulated model is

$$f_j\left(y_t|v_{1;t}, v_{2;t}, v_{3;r}\right)$$

  for components $j = 1, 2$.
- Rows 17–25 prepare initial statistics and parameters for the estimation.
- Rows 28–52 represent the time loop.
  - Rows 30–32 compute logarithmic proximities.
  - Rows 33–34 perform normalization of proximities and take their exponent.
  - Rows 36–42 do statistics update and construction of point estimates of the parameters.
  - Rows 44–52 construct the output zero-step prediction (either point one - expectation or generated one - expectation + noise).