```
// T52mixExpNor.sce
// MIXTURE ESTIMATION (descriptive normal)
// - static normal componens
// - multinomial y and v
//   |y1|    |a11 a12 a13||v1|
//   |  | =  |           ||v2| + e
//   |y2|    |a21 a22 a23||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 reg. coefficients  // 4
        3 1 2];                                                    // 5
thS(2)=[8 1 5                                                      // 6
        4 6 8];                                                    // 7
thS(3)=[6 9 1                                                      // 8
        5 3 2];                                                    // 9
```

```
alS=[.3 .2 .5];                       // parameters of pointer model
sd=.1;                                // common and known noise std.
for t=1:nd
  jS=sampCat(alS);                    // pointer value generation
  cS(t)=jS;
  v(:,t)=randn(1,3);                  // expl. variables
  y(:,t)=thS(jS)*v(:,t)+sd*randn(2,1); // output value generation
end
nc=length(thS);                       // number of componens

// INITIALIZATION
V=list(); thE=list();
ka=[1 1 1];                           // initial counter stat.
for j=1:nc
  thE(j)=thS(j)+5*rand(2,3);          // def. of initial parameters
  V(j)=eye(5,5);                      // initial inf. matrix
  V(j)(1:2,3:5)=thE(j);
  V(j)(3:5,1:2)=thE(j)';
  thE(j)=v2thN(V(j)/ka(j),2);         // initial point estimates
end

// TIME LOOP
for t=1:nd
```

```
// estimation
for j=1:nc
  [nill,qp(j)]=GaussN(y(:,t),thE(j)'*v(:,t),sd*eye(2,2)); //proximity
end
q=exp(qp-max(qp));
w=q/sum(q);                              // weights
wt(:,t)=w;
for j=1:nc
  Ps=[y(:,t)' v(:,t)']';                 // extended reg. vector
  V(j)=V(j)+w(j)*Ps*Ps';                 // update of statistics V
  ka(j)=ka(j)+w(j);                      // update of counter
  thE(j)=v2thN(V(j)/ka(j),2);            // point estimates
  c(j).th(:,t)=thE(j)(:);                // remember for plot
end

// zero-step prediction
ypp=0;
for j=1:nc
  select 1        // type of prediction: 1-point, 2-generated;
  case 1, ypp=ypp+w(j)*thE(j)'*v(:,t);
  case 2, ypp=ypp+w(j)*(thE(j)'*v(:,t)+sd*randn(2,1));
  end
end
```

```
  yp(:,t)=ypp;                          // prediction
end

// RESULTS
set(scf(1),'position',[600 10 1200 400])
for j=1:nc
  subplot(1,3,j)
  plot(c(j).th')
  title('Evolution of estimated parametrs: comp. '+string(j))
end

disp 'Simulated parameter values'
disp(thS(:))
disp 'Final estimated parameters'
disp(thE(:))

set(scf(2),'position',[900 500 600 400])  // plot data and prediction
plot(1:nd,y,'ob',1:nd,yp,'xr','markersize',8)
title 'Data (b) and prediction (r)'

[nill,cp]=max(wt,'r');                  // accuracy of classification
disp 'Accuracy of classification'
ACC=acc(cS,cp)
```

```
                                                                    // 79
disp 'Relative prediction error'          // accuracy of prediction  // 80
RPE=rpe(y(:),yp(:))                                                  // 81
```

Description of the program

- Rows 3–12 define parameters for simulation.
- Rows 12–17 perform simulation with switching components. Simulated model is multivariate.
- Rows 21–29 prepare initial statistics and parameters for the estimation.
- Rows 32–57 represent the time loop.
  - Rows 34–36 compute logarithmic proximities.
  - Rows 38–39 perform normalization of proximities and take their exponent.
  - Rows 40–46 do statistics update and construction of point estimates of the parameters.
  - Rows 49–55 construct the output zero-step prediction (either point one - expectation or generated one - expectation + noise).