```
// T42mixDesNor1.sce
// MIXTURE ESTIMATION (descriptive, normal)
// - static normal componens
// - two-dimensional variable
// Experiments
// - change simulated expectations thS
// - change initial expectations thE (through the statistics S, ka)
// ---------------------------------------------------------------------
exec("ScIntro.sce",-1),
getd(), mode(0)

nd=500;                                                          // 1
// PARAMETERS                                                    // 2
thS=[1 3 9 15];                        // simulated comp. expectations // 3
sd=[1.5 .8 1.2 .9]*1;                  // standard deviations          // 4
nc=length(sd);                         // number of components         // 5
alS=[.1 .2 .4 .3];                     // parameters of pointer model  // 6
                                                                 // 7
// SIMULATION                                                    // 8
for t=1:nd                                                       // 9
  jS=sampCat(alS);                     // pointer value                // 10
  cS(t)=jS;                            // stor pointer value           // 11
  y(t)=thS(jS)+sd(jS)*randn();         // output                       // 12
```

```
end                                                                    // 13
                                                                       // 14
// ESTIMATION                                                          // 15
// initialization                                                      // 16
thE=[0 5 10 15];                                                       // 17
ka=[1 1 1 1];                          // initial counter              // 18
V=list();                                                              // 19
for j=1:nc                                                             // 20
  V(j)=[1 thE(j); thE(j) 1];                                           // 21
  thE(j)=v2thN(V(j)/ka(j),1);          //initial point estimates       // 22
end                                                                    // 23
                                                                       // 24
// time loop of estimation                                             // 25
th=thE;                                                                // 26
for t=1:nd                                                             // 27
  for j=1:nc                                                           // 28
    [nill,q(j)]=GaussN(y(t),thE(j),.1);// log-proximity                // 29
  end                                                                  // 30
  qn=exp(q-max(q));                    // normalized proximity        // 31
  w=qn/sum(qn);                        // weights                     // 32
  wt(:,t)=w;                           // remember weights            // 33
  for j=1:nc                                                           // 34
    Ps=[y(t) 1];                       // extended regression vector  // 35
```

```
    V(j)=V(j)+w(j)*Ps'*Ps;              // update of inf. matrix        // 36
    ka(j)=ka(j)+w(j);                   // update of counter            // 37
    thE(j)=v2thN(V(j)/ka(j),1);         // point estimates              // 38
  end                                                                  // 39
  th=[th; thE];                         // remember point estimates     // 40
end                                                                    // 41
                                                                       // 42
// RESULTS                                                             // 43
tx=['b';'r';'g';'k'];                                                  // 44
set(scf(1),'position',[600 10 600 400])   // evolution of par. est.     // 45
for j=1:nc                                                             // 46
  plot(th(:,j),'-'+tx(j))                                              // 47
end                                                                    // 48
title 'Evolution of the estimated parameters'                          // 49
legend('c1','c2','c3','c4');                                           // 50
                                                                       // 51
disp 'The final parmeter estimates are'                                // 52
disp(thE)                                                              // 53
                                                                       // 54
[nill,cp]=max(wt,'r');                    // accuracy of classification // 55
disp 'Accuracy of classification'                                      // 56
ACC=acc(cS,cp)                                                         // 57
```

Description of the program

This program uses scalar normal components of the descriptive type

$$x_t = \theta_j + e_t$$

where θ_j are parameters (constants) of regression type components - j denoted the components.

The pointer model has static categorical distribution

$$f\left(c_t = j|\alpha\right) = \alpha_j$$

This pointer model is used for simulation, however, in estimation it is neglected due to its little importance for classification - only its stationary form (without the knowledge of the actual output value) is estimated.

- Rows 3–6 define parameters of the models
- Rows 9–13 perform data simulation (first the pointer value is generated and the the corresponding component is used for generating the output value).
- Rows 16–23 are devoted to initialization of the estimation
  - Row 17 defines the initial values of the regression coefficients.
  - Rows 18–23 construct corresponding initial statistics.
- Rows 26–41 perform the time loop for estimation.

– Rows 28–30 compute the proximities in a logarithmic form.

– Rows 31–32 perform exponentiation and normalization to weights w.

– Then, for each component, the statistics are updated and point estimates of the parameters are recomputed.