```
// T47mixDesBin.sce
// MIXTURE ESTIMATION (descriptive, modified binomial)
// Experiments
// - change simulated parameters
// - change initial parameters
// --------------------------------------------------------------------
exec("ScIntro.sce",-1),
getd(), mode(0)

nd=500;                                                              // 1
ni=1;                                                               // 2
// PARAMETERS                                                       // 3
N1=5;                                                              // 4
pS=[.2 .5 .9];                     // component parametrs          // 5
nc=length(pS);                     // number of components         // 6
alS=[.3 .3 .4];                    // parameters of pointer model  // 7
                                                                   // 8
// SIMULATION                                                      // 9
for t=1:nd                                                         // 10
  jS=sampCat(alS);                 // pointer value                // 11
  cS(t)=jS;                        // stor pointer value           // 12
  y(t)=sampBin1(pS(jS),N1);        // output generation            // 13
end                                                                // 14
```

```
                                                            // 15
// ESTIMATION                                               // 16
// initialization                                           // 17
ka=[1 1 1]*ni;                       // initial counter      // 18
pE=[.1 .3 .5];                       // initial parameters   // 19
S=(N1-1)*pE.*ka+ka;                  // statistics for given pE  // 20
                                                            // 21
// time loop                                                // 22
for t=1:nd                                                  // 23
  for j=1:nc                                                // 24
    q(j)=binom1(y(t),pE(j),N1);      // proximity            // 25
  end                                                       // 26
  w=q/sum(q);                        // weights              // 27
  wt(:,t)=w;                         // remember weights     // 28
  for j=1:nc                                                // 29
    S(j)=S(j)+w(j)*y(t);             // update of            // 30
    ka(j)=ka(j)+w(j);                //   statistics         // 31
    pE(j)=(S(j)-ka(j))/((N1-1)*ka(j)); // parameter estimates  // 32
    pEt(t,j)=pE(j);                  // remember             // 33
  end                                                       // 34
end                                                         // 35
                                                            // 36
// RESULTS                                                  // 37
```

```
tx=['b';'r';'g'];                                                  // 38
set(scf(1),'position',[600 10 600 800]) // evolution of par. est.  // 39
  title 'Evolution of the estimated parameters'                    // 40
for j=1:nc                                                         // 41
  subplot(3,1,j)                                                    // 42
  plot(pEt,'-')                                                     // 43
  xlabel('component '+string(j))                                    // 44
end                                                                // 45
                                                                   // 46
disp 'The final parmeter estimates are'                            // 47
disp(pE)                                                           // 48
                                                                   // 49
[nill,cp]=max(wt,'r');                  // accuracy of classification  // 50
disp 'Accuracy of classification'                                  // 51
ACC=acc(cS,cp)                                                     // 52
```

Description of the program

- Row 2 sets the strength of prior information (n_i is as if the number of prior data from which the information has been gained).

- Rows 4–7 define parameters of the task.

- Rows 10–14 perform simulation - generation from categorical distribution of individual components.

- Rows 16–20 prepare initialization of the task

  - Row 25 computes the statistics to be in accord with the values of initial the parameters

- Rows 23–35 run the time loop.

  - Row 25 computes the proximities.
  - Rows 27–28 construct the component weights.
  - Rows 29–34 update the statistics and recompute the values of the point estimates pf the parameters.