

```

// T46mixDesCat.sce
// MIXTURE ESTIMATION (descriptive, categorical)
// - static multivariate componens
// Experiments
// - change simulated parameters
// - change initial parameters
// -----
exec("ScIntro.sce",-1),
getd(), mode(0)

nd=500; // 1
// PARAMETERS // 2
//   x1  1  1  2  2 // 3
//   x2  1  2  1  2 // 4
c(1).pS=[.9 .02 .05 .03]; // 5
c(2).pS=[.05 .02 .9 .03]; // 6
c(3).pS=[.05 .9 .03 .02]; // 7
nc=length(c); // number of components // 8
alS=[.3 .3 .4]; // parameters of pointer model // 9
// 10
// SIMULATION // 11
for t=1:nd // 12
    jS=sampCat(alS); // pointer value // 13

```

```

    cS(t)=jS;                // stor pointer value           // 14
    z=sampCat(c(jS).pS);     // output generation       // 15
    y(t,:)=col2xt(z,[2 2]); // 16
end                           // 17
                             // 18

// ESTIMATION                // 19
// initialization            // 20
ka=[1 1 1];                 // initial counter     // 21
c(1).pE=[.7 .1 .1 .1];     // initial parmeters   // 22
c(2).pE=[.1 .1 .7 .1];     // 23
c(3).pE=[.1 .7 .1 .1];     // 24
for j=1:nc                  // 25
    c(j).S=ka(j)*c(j).pE;   // initial statistics  // 26
end                           // 27
                             // 28

// time loop of estimation   // 29
b=max(y,'r');               // numbs of values in vars // 30
for t=1:nd                  // 31
    for j=1:nc              // 32
        z=xt2col(y(t,:),b); // 33
        q(j)=c(j).pE(z);    // proximity           // 34
    end                     // 35
    w=q/sum(q);             // weights             // 36

```

```

    wt(:,t)=w;                                // remember weights           // 37
    for j=1:nc                                // 38
        c(j).S(z)=c(j).S(z)+w(j);             // update of statistics           // 39
        c(j).pE=fnorm(c(j).S);               // parametrs                      // 40
        c(j).th(t,:)=c(j).pE;                // remember                      // 41
    end                                       // 42
end                                         // 43
                                           // 44
// RESULTS                                // 45
tx=['b','r','g'];                          // 46
set(scf(1),'position',[600 10 600 800])    // evolution of par. est.        // 47
    title 'Evolution of the estimated parameters' // 48
for j=1:nc                                // 49
    subplot(3,1,j)                          // 50
    plot(c(j).th,'-' + tx(j))               // 51
    xlabel('component ' + string(j))         // 52
end                                         // 53
                                           // 54
disp 'The final parmeter estimates are'     // 55
for j=1:nc                                // 56
    c(j).pE                                  // 57
end                                         // 58
                                           // 59

```

```

[nill,cp]=max(wt,'r');           // accuracy of classification // 60
disp 'Accuracy of classification' // 61
ACC=acc(cS,cp)                   // 62

```

## Description of the program

- Rows 3–9 define parameters of the task.
- Rows 12–17 perform simulation - generation from categorical distribution of individual components.
- Rows 21–27 prepare initialization of the task
  - Row 26 computes the statistics to be in accord with the values of initial the parameters
- Row 30 sets the base vector - numbers of different values of variables. This is necessary for coding the variable vectors into one variable in row 33.
- Rows 34–44 run the time loop.
  - Rows 32–35 compute the proximities.
  - Rows 36–37 construct the component weights.
  - Rows 38–43 update the statistics and recompute the values of the point estimates of the parameters.