```
// T26preCat.sce
// PREDICTION WITH DISCRETE MODEL (ON-LINE)
// Change: - length of prediction
//         - uncertainty of the simulated model
//         - imput signal
//         - study the beginning when estimation is not finished
//           how can we secure quicker transient phase of estimation?
// Remark: another way og generation is
//            y(t)=sum(rand(1,1,’u’)>cumsum(th(i,:)))+1;
// -------------------------------------------------------------------
exec("ScIntro.sce",-1), mode(0)

nd=500;                            // number of data                    // 1
// PARAMETRS                                                            // 2
np=2;                              // length of prediction              // 3
th1=[0.98 0.01 0.04 0.97];         // parameters (for y=1)              // 4
thS=[th1; 1-th1];                  // full parameter                    // 5
u=(rand(1,nd+np,’u’)>.3)+1;        // input generation                  // 6
y(1)=1;                            //initial y                          // 7
b=[2 2];                           // base (numb. of values of u and y) // 8
thE=genProb(2,4,1);                // initial setting of parameter      // 9
                                                                        // 10
S=1e-8*ones(2,4);                  // initial statistics                // 11
```

```
Et=zeros(4,nd-np);
// TIME LOOP
for t=2:nd
  // prediction
  k=xt2col([u(t) y(t-1)],b);       // column of the model table
  yy=sampCat(thE(:,k));            // first step of prediction
  for j=1:np
    k=xt2col([u(t) yy],b);         // column of the model table
    yy=sampCat(thE(:,k));          // further steps of prediction
  end
  yp(t+np)=yy;                     // np-step predction

  // simulation
  k=xt2col([u(t) y(t-1)],b);       // column in model table
  y(t)=sampCat(thS(:,k));          // generation of discrete var.

  // estimation
  k=xt2col([u(t) y(t-1)],[2 2]);   // row of model matrix
  S(y(t),k)=S(y(t),k)+1;           // statistics update
  thE=fnorm(S,1);                  // pt estimates
  Et(:,t)=thE(1,:)’;
end
```

```
// RESULTS                                                              // 35
disp(' Simulated parameters')                                           // 36
disp(thS)                                                               // 37
disp(' Estimated parameters')                                           // 38
disp(thE)                                                               // 39
                                                                        // 40
s=np+2:np+51;                                                           // 41
set(scf(),'position',[500 100 1000 400])                                // 42
subplot(121),plot(Et')                                                  // 43
title('Evolution of estimated parameters')                              // 44
set(gca(),"data_bounds",[0 nd-np+1 -.1 1.1])                            // 45
subplot(122),plot(s,y(s),s,yp(s),'.:')                                  // 46
title('First 50 outputs and their prediction')                          // 47
set(gca(),"data_bounds",[s(1) s($) .9 2.1])                             // 48
                                                                        // 49
s=np+2:nd;                                                              // 50
Wrong=sum(y(s)~=yp(s))                                                  // 51
From=nd-np                                                              // 52
```

Description of the program

XXXXX

Rows 3–5 prepare values of parameters

Rows 9–13 prepare initial statistics and parameters

Rows 15–31 perform the time loop

16: generation of the pointer value
17: is simulation of the output
22: computes proximities
23: weights normalized proximities
27–28: update of statistics
29: point estimates of parameters