

Odhad směsi s dynamickým ukazovátkem a diskrétními komponentami

- diskrétní výstup - řízená koruna s pamětí $f(y_t|u_t, y_{t-1})$
- simulovaná data
- inicializace odhadu - zašuměné parametry ze simulace
- model ukazovátka $f(c_t|c_{t-1}, \alpha) = \alpha_{c_t|c_{t-1}}$

Simulují se data ze směsi kategorických komponent a dynamického ukazovátka:

- komponenta i pro $i = 1, 2, \dots, n_c$

$$\begin{array}{c|cc}
 u_t, y_{t-1} & y_t = 1 & y_t = 2 \\
 \hline
 1, 1 & \Theta_{1|11}^i & \Theta_{2|11}^i \\
 1, 2 & \Theta_{1|12}^i & \Theta_{2|12}^i \\
 2, 1 & \Theta_{1|21}^i & \Theta_{2|21}^i \\
 2, 2 & \Theta_{1|22}^i & \Theta_{2|22}^i
 \end{array} \quad (0.1)$$

kde y_t a u_t jsou výstup a vstup v čase t ,

- ukazovátko

$$\begin{array}{cccc}
 & c_t = 1 & c_t = 2 & \dots & c_t = n_c \\
 c_{t-1} = 1 & \alpha_{1|1} & \alpha_{2|1} & & \alpha_{n_c|1} \\
 c_{t-1} = 2 & \alpha_{1|2} & \alpha_{2|2} & & \alpha_{n_c|2} \\
 \dots & & & \dots & \\
 c_{t-1} = n_c & \alpha_{1|n_c} & \alpha_{2|n_c} & & \alpha_{n_c|n_c}
 \end{array}$$

n_c je počet komponent.

Předpoklady:

Sci značení:

Úloha: Simulace a odhad dynamickým modelem směsi diskrétních komponent - základní konfigurace pro odhad směsi s diskrétními komponentami.

Poznámky

1. *Dynamické ukazovátko umožňuje predikci aktivní komponenty, protože dává do souvislosti po sobě jdoucí aktivity komponent.*
2. *Inicializace odhadu směsi komponent je velice důležitá. Bez dobré inicializace může být konvergence v odhadu pomalá nebo odhad selže.*

Vzorec pro váhy komponent, které přiřazují komponentám pravděpodobnosti aktivity, se sestává ze tří částí. Jsou to (i) vektor "vzdáleností" aktuálně změřeného datového vzorku od jednotlivých komponent, (ii) model ukazovátka a (iii) minulý váhový vektor w_{t-1} . Druhá a

třetí položka jsou stejné, jako u směsi spojitých komponent. Vzdálenost změřeného datového vzorku od komponenty je dán hodnotou modelu s dosazeným aktuálním odhadem parametrů a hodnotami změřených dat. Model v diskrétním případě je dán tabulkou (0.1). Prvky této tabulky jsou bodové odhady parametrů a změřená data y_t , u_t a y_{t-1} určují jeden prvek této tabulky. A hodnota tohoto prvku určuje zjišťovanou "vzdálenost".

Z uvedeného plyne, že komponenty, které mají některý prvek stejný, mají od příslušného datového vzorku stejnou vzdálenost.

Zde jako počáteční hodnoty parametrů pro odhad použijeme skutečné hodnoty parametrů, a ty více nebo méně zašumíme.

Program

Popis programu

1. Simulace hodnot ukazovátka a podle toho generování dat z příslušné komponenty.
2. Odhad se provádí podle standardních vzorců:
 - (a) výpočet vah W_t a w_t pro změřený výstup y_t .
 - (b) Přepočítání statistik komponent a ukazovátka.
 - (c) Konstrukce bodových odhadů parametrů.
3. Jako výsledek se ukazují hodnoty odhadovaného ukazovátka ve srovnání se simulovaným. Tím úloha získává také charakter klasifikace.

Kód programu

```
// P72MixCat.sce
// Mixture estimation
// - discrete components
// - dynamic pointer model
// - simulated data
// - initialization by parameters from simulation + noise
// Rem: distinguishing of discrete models is rather difficult
[u,t,n]=file();           // find working directory
chdir(dirname(n(2)));     // set working directory
clear("u","t","n")      // clear auxiliary data
exec("ScIntro.sce",-1),mode(0) // intro to session
rand('seed',0)

nd=100;                   // number of data
nc=3;                     // number of component
I_estCov=0;              // estimation of noise covariances ? 0|1 no|yes

// simulated reg.coef.
a=.01;                   // amplitude of noise in comp. params
Sim.nc=nc;
```

```

Sim.Cz(1).th=fnorm([1 0; 0 1; 1 0; 0 1]+a*rand(4,2,'u'),2);
Sim.Cz(2).th=fnorm([0 1; 1 0; 0 1; 1 0]+a*rand(4,2,'u'),2);
Sim.Cz(3).th=fnorm([1 0; 1 0; 0 1; 0 1]+a*rand(4,2,'u'),2);
thu=[.4 .6];
// simulated pointer parameterer
a=.1; // amplitude of noise in ptr. params
al=[1 0 0; 0 1 0; 0 0 1]; // main dynamic part of ptr.par
Sim.Cp.th=fnorm(al(1:nc,1:nc)+a*rand(nc,nc,'u'),2); // + noise

// initial parameters
a=.1; // ratio of noise in init.pars
for j=1:nc // from those used in simulation
    [mr,mc]=size(Sim.Cz(j).th);
    Est.Cz(j).V=10*(Sim.Cz(j).th+a*rand(mr,mc,'u')); // statistics
    Est.Cz(j).th=fnorm(Est.Cz(j).V,2); // par. point est.
end
Est.Cp.V=.1*ones(nc,nc); // pointer statistics
Est.Cp.th=fnorm(rand(nc,nc,'u')+1); // pointer parameter
w=fnorm(ones(1,nc)); // weights

Sim.ct(1)=1; // initial pointer
Sim.yt(1)=1; // initial output

// SIMULATION =====
for t=2:nd
    u=sum(rand(1,1,'u')>cumsum(thu))+1;
    j=Sim.ct(t-1); // last active component
    Sim.ct(t)=sum(rand(1,1,'u')>cumsum(Sim.Cp.th(j,:)))+1; // pointer
    i=(length(thu)-1)*u+Sim.yt(t-1); // actual row in model
    k=Sim.ct(t); // actual active component
    y=sum(rand(1,1,'u')>cumsum(Sim.Cz(k).th(i,:)))+1; // output
    Sim.yt(t)=y; // stor
    Sim.ut(t)=u;
end

// ESTIMATION =====
tt=fix(nd/10); printf(' ')
for t=2:nd
    if t/tt==fix(t/tt), printf(' '); end
    for j=1:nc
        i=(max(Sim.ut)-1)*Sim.ut(t)+Sim.yt(t-1);
        q(j)=Est.Cz(j).th(i,Sim.yt(t)); // likelihood
    end

    ww=(q*w)'.*Est.Cp.th; // matrix weights
    W=ww/sum(ww); // normalization - f(c(t),c(t-1)|d(t))
    w=sum(W,1); // marginalization - f(c(t)|d(t))
    wt(:,t)=w'; // stor

```

```

// Update of statistic
Est.Cp.V=Est.Cp.V+W; // ptr.stat. update
for i=1:nc
    j=(max(Sim.ut)-1)*Sim.ut(t)+Sim.yt(t-1);
    D=zeros(nr,mc); D(j,Sim.yt(t))=1;
    Est.Cz(i).V=Est.Cz(i).V+w(i)*D; // pointer statistics

    // estimates of parametrs
    Est.Cz(i).th=fnorm(Est.Cz(i).V,2);
    Est.Cz(i).tht(:,t)=Est.Cz(i).th(:); // store estimates
end
Est.Cp.th=fnorm(Est.Cp.V,2); // pointer values
[ss,Est.ct(1,t)]=max(w); // store pointer
end

// Results
disp(Sim.Cp.th,'pt.pars_sim')
disp(Est.Cp.th,'pt.pars_est')

s=2:nd;
wr=sum(Sim.ct(s)~=Est.ct(s)');
printf('\n Wrong classifications %d from %d\n',wr,length(s))

set(gcf(1),'position',[50 50 400 400])
plot(s,Sim.ct(s),'bo',s,Est.ct(s),'rx')
title('Evaluation of weights w')
set(gca(),'data_bounds',[min(s) max(s) .9 nc+.1])

set(gcf(2),'position',[550 50 400 600])
for i=1:nc
    subplot(nc,1,i)
    plot(Est.Cz(i).tht')
    title('Evolution of parameter estimates'+string(i))
end
end

```