

## Odhad směsi s exponenciálními komponentami

Charakteristikou exponenciálního rozdělení je nezápornost a ne příliš strmý pokles pro rostoucí hodnoty.

Pro modelování komponent se budeme zabývat jednorozměrným rozdělením (jedna veličina). Vícerozměrné budeme brát pouze pro nezávislé veličiny, a tedy, hustotou pravděpodobnosti bude součin jednorozměrných. Komponenty se závislými veličinami zatím neumíme.

Hustota pravděpodobnosti

$$f(y) = \frac{1}{d} \exp\left\{-\frac{y}{d}\right\} \quad (1)$$

kde  $d$  je střední hodnota rozdělení.

Rozptyl je  $1/d^2$

Věrohodnostní funkce (součin modelů) je

$$L_N(d) = \prod_{t=1}^N \frac{1}{d} \exp\left\{-\frac{y_t}{d}\right\} = \left(\frac{1}{d}\right)^{\kappa_N} \exp\left\{-\frac{S_N}{d}\right\}$$

kde  $S_N = \sum_{t=1}^N y_t$  a  $\kappa_N = N$ .

Přepočet statistik je

$$S_t = S_{t-1} + y_t, \quad \kappa_t = \kappa_{t-1} + 1 \quad (2)$$

Bodový odhad parametru  $d$  je

$$\hat{d}_N = \frac{S_N}{\kappa_N} = \bar{y} \quad (\text{průměr}) \quad (3)$$

**Odhad směsi** v každém čase probíhá podle následujícího schématu

1. Výpočet vah  $w$  pro jednotlivé komponenty, kde hlavním prvkem jsou tzv. proximity - tj. hodnoty hustoty pravděpodobnosti příslušné komponenty s dosazenými existujícími bodovými odhady parametrů a aktuálně změřenými daty, tj.  $f_c(y_t | \hat{\Theta}_{c;t-1})$ , kde  $c = 1, 2, \dots, nc$  (pro všechny komponenty).
2. Přepočet statistik, kde data se přidávají s příslušnou vahou.
3. Výpočet bodových odhadů.

Pro tyto kroky potřebujeme

1. Model **1**.
2. Vzorec pro přepočet statistik **2**.
3. Vzorec pro výpočet bodového odhadu parametrů **3**

Všechny tyto kroky jsou specifické pro určité rozdělení (zde exponenciální).

### Poznámka

*Proximity, počítané na hustotě pravděpodobnosti modelu nejsou příliš výrazné. Lze jmenovat několik důvodů: (i) Exponenciála klesá poměrně pomalu a její rozptyl je pevně svázán se střední hodnotou  $E[y] = d$ ,  $D[y] = d^2$ . Její pokles je proto dán střední hodnotou, která je většinou pro rozlišení komponent rozhodující, a nelze ho uměle zrychlit. (ii) Nejdůležitější část exponenciálního rozdělení co do počtu realizací je v počátku. A právě počátek je společný všem komponentám.*

*Proto je lepší, i když nesystematické, použít pro výpočet proximity nějakou jinou funkci, která nemá jmenované nevýhody. Jako asi nejlepší se jeví hustota pravděpodobnosti normálního rozdělení se stejnou střední hodnotou a rozptylem jako bylo odhadnuto pro exponenciální rozdělení.*

**Program** (komentáře jsou za programem)

```
// Pokusy s exp proximity jako likelihood
// - 3 komponenty
// - 2D data
// -----
exec SCIHOME/ScIntro.sce, mode(0)

nix=10;
Sim.Cx(1).th=[.5 5]';
Sim.Cx(2).th=[6 .5]';
Sim.Cx(3).th=[13 15]';
Est.Cpx.th=[.5 .2 .3];
Est.Cpx.V=.1*ones(1,3);           // pointer statistics
Est.kax=nix*ones(1,3);           // counter
cx(1)=1;

nd=1000;
for t=2:nd
    cx(t)=sum(rand(1,1,'u')>cumsum(Est.Cpx.th))+1;
    u=rand(2,1,'u'); a=Sim.Cx(cx(t)).th;
    x(:,t)=-log(u).*a;
end

Vx=[Sim.Cx(1).th Sim.Cx(2).th Sim.Cx(3).th];
for i=1:3
    Est.Cx(i).V=Vx(:,i)*nix;
    Est.Cx(i).th=Est.Cx(i).V;
end

printf(' ') // ===== TIME LOOP =====
for t=2:nd
    if t/fix(nd/10)==fix(t/fix(nd/10)), printf(' '); end

    // váhy 1. ÚSEK
    for j=1:3
        [xxx,Gx(j)]=ExpN(x(:,t),Est.Cx(j).th); // model
```

```

// [xxx,Gx(j)]=ExpMLik(x(:,t),Est.Cx(j).V,Est.kax(j)); // likelihood
// [xxx,Gx(j)]=GaussN(x(:,t),Est.Cx(j).th,.01*eye(2,2));
// dx=x(:,t)-Est.Cx(j).th; dx=dx(:);
// Gx(j)=1/(dx'*dx)**5;
end

Lx=Gx-max(Gx); // rough normlization
qx=exp(Lx); // exponent

ww=qx'*Est.Cpx.th; // matrix weights
wx=ww/sum(ww); // normalization - f(c(t),c(t-1)|d(t))
wtx(:,t)=wx'; // stor
//xx=x(:,t)', wx,cxx=cx(t)
//if t>10, pause, end
// Update of statistic 1ST COMPONENT =====
Est.kax=Est.kax+wx; // counter
Est.Cpx.V=Est.Cpx.V+wx; // ptr.stat. update
Psx=[x(:,t)' 1]; // extended reg.vec.
for i=1:3
// přepoččet statistiky
Est.Cx(i).V = Est.Cx(i).V + wx(i)*x(:,t); // information matrix

//bodové odhady
Est.Cx(i).th=Est.Cx(i).V/(ones(2,1)*Est.kax(i));
Est.Cpx.th=fnorm(Est.Cpx.V);
end
end
disp ' '
[xxx,Ecx]=max(wtx,'r');

//USAGE: ct=q(Ect)
// set: [q T]=c2c(ct,Ect); simul and estim pointer
// plot(1:nd,ct,1:nd,q(Ect)) plotting

[q,T]=c2c(cx',Ecx);
Ecc=q(Ecx);
wrong=sum(cx'~=Ecc')
from=nd

nc=3;
// sim -----
CS=list();
for i=1:nc
j=find(cx==i);
CS(i)=x(:,j);
end

tx=['b.';'r.';'g.'];
scf(1);

```

```

for i=1:nc
    plot(CS(i)(1,:),CS(i)(2,:),tx(i),'markersize',3)
end
title 'Simulated clusters'
set(gca(),'data_bounds',[0 45 0 40])

// est -----
C=list();
for i=1:nc
    j=find(Ecc==i);
    C(i)=x(:,j);
end

set(gcf(2),'figure_position',[850 200]);
for i=1:nc
    plot(C(i)(1,:),C(i)(2,:),tx(i),'markersize',3)
end
title 'Estimated clusters'
set(gca(),'data_bounds',[0 45 0 40])

```

Inicializace - dodělat !!!!