Odhad směsi s rovnoměrnými komponentami

Hlavní charakteristikou rovnoměrného rozdělení je:

1. Modelování úplné neurčitosti v rámci daných mezí.
2. Ostré ohraničení oblasti přípustných hodnot náhodné veličiny.

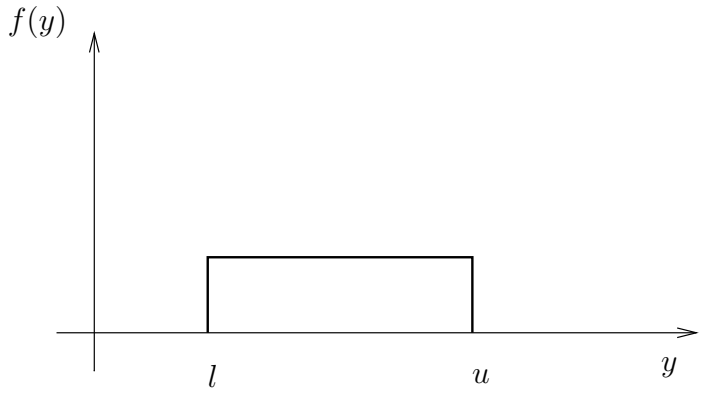
V této úloze chceme využít právě ostré vymezení modelovaných dat ve formě klastrů.

Tyto vyjímečné vlastnosti rovnoměrného rozdělení ale mají za důsledek to, že nepatří do exponenciální třídy rozdělení. To znamená, že nemá uzavřený rekurzivní algoritmus odhadování. Musí se proto používat různé heuristické postupy. Dále budeme demonstrovat proceduru odhadu použitou v programu.

Model

$$f\left(y|l,u\right)=\begin{cases}\frac{1}{u-l} & \text{pro } y\in\left(l,\,u\right)\\ 0 & \text{jinde}\end{cases}$$

tak, jak je to naznačeno na obrázku



Z tohoto obrázku je patrné, že pokud má veličina y toto rozdělení, pak nemůžeme dostat hodnoty menší než l a větší než u. Z hlediska odhadu je ale situace jiná. Pokud dostaneme hodnotu menší než l, je jasné, že veličina y nemůže mít rozdělení podle obrázku a parametr rozdělení l je nutno posunout doleva tak, aby změřená hodnota byla v intervalu $(l,\,u)$. Stejně je to i doprava. A právě na tomto principu je založen odhad rovnoměrného rozdělení.

Odhad parametrů modelu

Nicméně začneme poctivě od věrohodnostní funkce $L_N\left(\Theta\right)$, kde $\Theta=\left\{l,\,u\right\}$. Budeme se věnovat jednorozměrnému problému. Vícerozměrný budeme uvažovat pro nezávislé veličiny. Pro závislé je situace příliš komplikovaná.

$$L_N\left(\Theta\right)=\prod_{t=1}^{N}f\left(y_t|l,u\right)=\prod_{t=1}^{N}\frac{1}{u-l}\chi\left(y_t\geq l\right)\chi\left(y_t\leq u\right)=$$

$$= \left(\frac{1}{u-l}\right)^N \prod_{t=1}^{N} \chi\left(y_t \geq l\right) \chi\left(y_t \leq u\right)$$

kde $\chi(\cdot)$ je indikátorová funkce, která je rovna 1, když je podmínka splněna a 0 když není.

Dále si všimneme podmínek v indikátorových funkcích

$$y_1 \geq l \wedge y_1 \leq u$$
$$y_2 \geq l \wedge y_2 \leq u$$
$$\ldots$$
$$y_N \geq l \wedge y_N \leq u$$

Z nich plyne

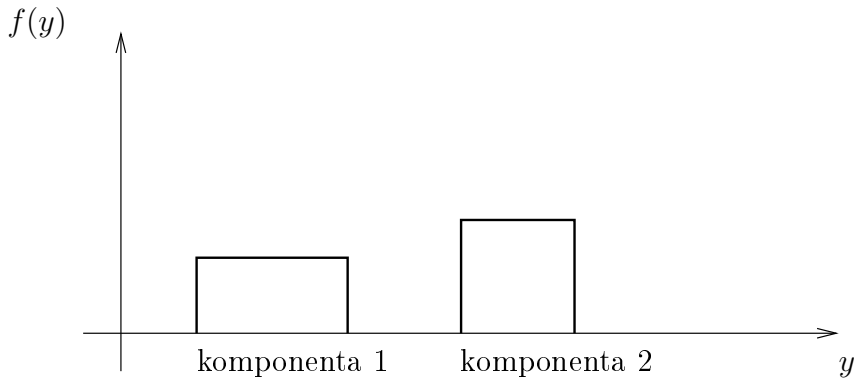
$$l \leq \min\left(y_t\right) \wedge u \geq \max\left(y_t\right)$$

tj. parametr l musí být menší, než nejmenší naměřené y_t a u vetší, než největší y_t. Přitom ale hodnota věrohodnostní funkce je $\left(\frac{1}{u-l}\right)^N$. Tato hodnota bude tím větší, čím kratší bude interval (l, u) - to tlačí oba parametry k sobě. Odtud optimální odhad parametrů bude

$$\hat{l} = \min\left(y_t\right) \quad \text{a} \quad \hat{u} = \max\left(y_t\right)$$

Odhad směsi s rovnoměrnými komponentami

Jakmile chceme použít rovnoměrné modely jako komponenty modelu směsi distribucí, dostáváme se do potíží. Představme si nejjednodušší případ - jednorozměrnou směs s dvěma komponentami (podle obrázku)



Je jasné, že data budou přicházet z obou komponent. Pokud bychom použili odvozený princip odhadu, budou po pár měřeních obě odhadované komponenty stejné a budou překrývat všechna data.

Dobře, to jsme ale nerespektovali váhy komponent. Výpočet vah se zakládá především na výpočtu proximity - to je hodnota dosud odhadnuté komponenty s dosazeným změřeným y_t. Přičemž odhad komponenty se provádí výše uvedeným způsobem - padne li y_t mimo hustotu pravděpodobnosti, posuneme příslušnou hranici. Padne-li ale hodnota mimo, je hp rovna nule a tím i váha je nulová a k žádnému odhadu nedojde.

Proto je třeba výpočet vah opřít o jinou hustotu pravděpodobnosti, než rovnoměrnou. Nejlépe o normální rozdělení se stejnými momenty, jaké má dosud odhadnutá komponenta. Tak opět dostáváme "rozumné" váhy a můžeme každou komponentu odhadovat vždy s její váhou. To znamená, že u komponenty s malou váhou posuneme mez málo a naopak. Nicméně se vždy posouvají všechny meze, což opět, sice pomalu, ale přece, táhne komponenty na jedno místo uprostřed dat. Proto použijeme cosi jako zapomínání - délka intervalu posunutí mezí se dělí počtem dat příslušejících komponentě. Tím se posouvání mezí zpomaluje s přibývajícím množstvím zpracovaných dat (podobně jako třeba u exponenciálního zapomínání).

Algoritmus odhadu je dobře patrný z přiloženého programu (popis programu je na konci za výpisem)

```
// P74MixUnif.sce
// Mixture estimation with uniform components
//  Data:  cons. speed. gas. moment revs. slope
// ----------------------------------------
[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 sesion

function unifplot(th,s)
  // plot of a rectangle
  // th   [L1 U1
  //       L2 U2]
  // s    format of plot

  if argn(2)<2, s='b'; end

  for j=1:max(size(th))
    L1=th(j)(1,1);
    R1=th(j)(1,2);
    L2=th(j)(2,1);
    R2=th(j)(2,2);

    plot([L1,R1],[L2,L2],s)
    plot([L1,R1],[R2,R2],s)
    plot([L1,L1],[L2,R2],s)
    plot([R1,R1],[L2,R2],s)
  end
endfunction

// Real data: individual variables (vith denotation of their meaning)
// 1  SPOTREBA      y1         9  PRIC-ZRYCH    v3        y - output
// 2  SPEED (2+3)/2 y2        10  STAC-SPEED    v4        u - control
// 3  VOLANT        u1        11  ROAD-CAN      v5        v - ext. input
// 4  PLYN          u2        12  CAS-UTC       v6
// 5  BRZDA         u3        13  UTM-X2        v7
// 6  R-STUP-A      u4        14  UTM-Y2        v8
// 7  MOMENT-MOT    v1        15  ALTITUDE2     v9
```

```
// 8  OTACKY-MOT    v2        16  COURSE       v10
// ------------------------------------------------------------------------------

nd=6000;                      // number of data (max 75420/5 = 15084)

// DATA LOAD
load _data/dataKF.dat
dt=DataKF.dataOrig;           // data from driven car
dt=dt([2 7],1:5:$);           // each 5th sample is taken
                              // selected data are: speed and moment of engine
// INITIALIZATION
y=dt(:,1:nd);
th=list();                    // initial parameters (lower and upper bounds)
th(1)=[60 70
       50 100];
th(2)=[60 70
       0 5];
th(3)=[60 70
       -22 -18];
th(4)=[130 140
       100 110];
th(5)=[120 130
       0 5];
thI=th;                       // remember initial setting
ny=size(th(1),1);             // dimension of data (given by th)
nc=length(th);                // number of components (given by th)
nu=ones(1,nc);                // pointer statistics
al=fnorm(nu);                 // pointer parameter
m=zeros(1,nc);
Lm=zeros(1,nc);
E=list(); D=list();
thL=list(); thU=list();       // upper and lower borders
for j=1:nc
  thL(j)=zeros(ny,nd);
  thU(j)=zeros(ny,nd);
end
if ny~=size(dt,1), disp('ERROR: Wrong number of variables'), return, end // check

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

  // weights
  for j=1:nc
    E(j)=(th(j)(:,2)+th(j)(:,1))/2;          // expectation
    D(j)=(th(j)(:,2)-th(j)(:,1))**2/12;      // variance
    [xxx,Lm(j)]=GaussN(y(:,t),E(j),diag(D(j)));
  end
  Lm=Lm-max(Lm);
```

```
  m=exp(Lm);
  if sum(m)<-1E10, m=ones(1,nc); end
  w=m/sum(m);
  wt(:,t)=w';

  // on-line estimation - statistics
  if 0, nn=ones(1,nc); else nn=nu; end  // forgetting: yes or not
  for j=1:nc
    for i=1:ny
      yL=y(i,t)-th(j)(i,1);
      if yL<=0, th(j)(i,1)=th(j)(i,1)+w(j)*yL/nn(j); end
      yU=y(i,t)-th(j)(i,2);
      if yU>=0, th(j)(i,2)=th(j)(i,2)+w(j)*yU/nn(j); end
    end
  end

  // on-line estimation - parameters
  for j=1:nc
    thL(j)(:,t)=th(j)(:,1);
    thU(j)(:,t)=th(j)(:,2);
  end
  nu=nu+w;                                // pointer
  al=fnorm(nu);                           //   statistics and parameters
end

// RESULTS
set(scf(2),'position',[600 100 800 600])
for j=1:nc
  plot(y(1,:),y(2,:),'c.')    // data
end
unifplot(thI,'.:r')           // initial parameters
unifplot(th)                  // estimated parameters
title 'Uniform clusters (red: initial support, blue: estimated support)'
```

Popis programu

V programu se používají reálná data změřená na jedoucím automobilu, konkrétně rychlost a moment motoru. Počáteční hodnoty parametrů (meze) jsou určeny expertně z datových klastrů - získáme snadno pomocí funkce scatt().

V úvodu programu je seznam veličin, které jsou k dispozici pro testování. Můžeme zkoušet libovolné dvojice (nebo i vice veličin, pak jsou ale problémy se zobrazením) a také je potřeba upravit počáteční parametry.

Jinak je program poměrně srozumitelný.