Odhad hierarchické směsi

Hierarchická směs je přesně to, co se pod pojmem "hierarchická" myslí. Nahoře je směs s několika komponentami. Každá komponenta je opět směsí a může ukazovat na další směsi jako své komponenty.

Hlavní význam takového modelu je zřejmě v tom, že celý problém (klastrování a klasifikace) je na nejvyšší úrovni zhruba rozdělen do několika částí (komponenty vrchní směsi). Každá z těchto částí je potom dále podrobněji zkoumána (dělena) podle dalších podrobností v komponentách příslušné sub-komponenty z vyšší části. Pro lepší představu uvedeme konkrétní příklad.

Příklad

Sledujeme jízdu osobního automobilu a rádi bychom detekovali, zda jde o jízdu ekonomickou (opatrnou) nebo sportovní (divokou). To, jak se jízda projevuje se nejlépe pozná z měřeného příčného a podélného zrychlení. Ekonomická jízda má tato zrychlení většinou nulové, jen vyjímečně se objevují malé odchylky. Divoká jízda naopak má tato zrychlení většinou nenulová, občas mohou klesnout směrem k nule ale občas také mohou vyletět do větších hodnot.

Ale to, jak se změny v akceleraci projeví bude zřejmě záviset na tom, kde se automobil zrovna pohybuje. Vytipujeme si 3 druhy prostředí: ve městě, mimo město a na dálnici.

Vrchní komponenta modelu se bude týkat prostředí. Bude mít 3 komponenty (podle typů prostředí) a jako data zvolíme rychlost (do 50 km/h město, mezi 50 a 100 km/k mimo město a nad 100 km/h dálnice - tyto hodnoty využijeme pro apriorní nastavení horní směsi).

Každá z právě jmenovaných komponent bude zase směsí s dvěma komponentami. Jedna pro ekonomickou jízdu (malá zrychlení) a druhá pro divokou jízdu (větší zrychlení).

Model

Obecná formulace hierarchického modelu směsi plyne z uvedeného příkladu. Pro jednoduchost se ale budeme i nadále příkladu držet. Nejprve zavedeme značení:

Horní směs

y_t jsou jsou data pro horní směs (rychlost)

c_t je ukazovátko pro horní směs

Θ jsou parametry horních komponent (závisí na c_t)

α je parametr horního ukazovátka

Dolní směsi

d_t jsou data pro dolní směsi (příčné a podélné zrychlení)

k_t je ukazovátko pro dolní směsi (k_t bude závislé na c_t)

ϑ jsou parametry dolních komponent (závisí na c_t i k_t)

β jsou parametry dolních ukazovátek

Model pak bude

$$f\left(y_t, d_t | c_t, k_t, \Theta, \vartheta, \alpha, \beta\right) = f\left(y_t | c_t, \Theta, \alpha\right) f\left(d_t | k_t, c_t, \vartheta, \beta\right)$$

kde první část na pravé straně je model horní směsi a druhá část jsou modely dolních směsí.

Odhad parametrů modelu

Pro odvození odhadu vyjdeme opět ze sdružené pravděpodobnosti $\mathcal{J}$ (kde $D_{t-1} = [y(t-1), d(t-1)]$ je množina všech starých dat)

$$\mathcal{J} = f(y_t, d_t, c_t, k_t, \Theta, \vartheta, \alpha, \beta | D_{t-1}) =$$
$$= \underbrace{f(y_t|c_t, \Theta) f(\Theta|y(t-1))}_{\text{horní komponenty}} \times \underbrace{f(c_t|\alpha) f(c_{t-1}|y(t-1)) f(\alpha|y(t-1))}_{\text{horní pointer model}} \times$$
$$\times \underbrace{f(d_t|c_t, k_t, \vartheta) f(\vartheta|d(t-1))}_{\text{dolní komponenty}} \times \underbrace{f(k_t|c_t, \beta) f(\beta|d(t-1))}_{\text{dolní pointer model}}.$$

Označíme:

– predikce z horních komponent

$$p_{c_t} = \int_{\Theta^*} f(y_t|c_t, \Theta) f(\Theta|y(t-1)) d\Theta \doteq f\left(y_t|c_t, \hat{\Theta}_{t-1}\right)$$

– predikce z dolních komponent

$$q_{k_t|c_t} = \int_{\vartheta^*} f(d_t|k_t, c_t, \vartheta) f(\vartheta|d(t-1)) d\vartheta \doteq f\left(d_t|k_t, c_t, \hat{\vartheta}_{t-1}\right)$$

– predikce horního pointru

$$\hat{\alpha}_{c_t|c_{t-1}} = \int_{\alpha^*} f(c_t|c_{t-1}, \alpha) f(\alpha|y(t-1)) d\alpha$$

– predikce dolního pointru

$$\hat{\beta}_{k_t|c_t} = \int_{\beta^*} f(k_t|c_t, \beta) f(\beta|d(t-1)) d\beta$$

– staré váhy

$$f_{c_{t-1}} = f(c_{t-1}|y(t-1))$$

Sdruženou predikci všech ukazovátek můžeme psát

$$\mathcal{P} = f(c_t, c_{t-1}, k_t|y(t), d(t)) \propto p_{c_t} \hat{\alpha}_{c_t|c_{t-1}} f_{c_{t-1}} \times q_{k_t|c_t} \hat{\beta}_{k_t|c_t}$$

kde prvá část (do $\times$) patří horní směsi a druhá část dolním směsím.

Sdružená pravděpodobnost c_t a c_{t-1} je

$$W_t = f(c_t, c_{t-1}|y(t)) = \sum_{k_t} \mathcal{P},$$

pravděpodobnost c_t je marginála W_t

$$w_{u;t} = \sum_{c_{t-1}} W_t$$

a pravděpodobnost k_t je

$$w_{l;t} = \sum_{c_t} \sum_{c_{t-1}} \mathcal{P}.$$

Tím je určen výpočet potřebných vah.

Přepočet statistik

Přepočet statistik pro ukazovátka je standardní přepočet pro diskrétní model s příslušnými vahami komponent.

Přepočet statistik pro komponenty závisí na modelu, který se pro ně použije. Zde se opřeme o již zmíněný příklad o sledování jízdy osobního automobilu.

Horní komponenty pracují s veličinou rychlost a očekáváme centra komponent v hodnotách 40 (město), 80 (mino město) a 110 (dálnice). I když se jedná o nezápornou veličinu, můžeme pro ní uvažovat normální model (data budou přibližně symetrická podle zmíněných center).

Dolní komponenty modelují zrychlení. Protože očekáváme, že rozjezdů bude přibližně stejně jako brzdění a zatáček doleva tolik, kolik zatáček doprava, čekáme, že data budou symetrická kolem počátku - přičemž maximální počet hodnot pro ekonomickou jízdu bude v nule, pro sportovní jízdu kolem nuly. Klastry tedy budou v nule (ekonomická jízda) a jakýsi prstenec kolem nuly (sportovní jízda). Takové klastry ale nejsou vhodné pro detekci pomocí hustot pravděpodobností. Proto tyto data převedeme na absolutní hodnoty a dále s nimi budeme počítat v této formě.

Máme tedy nezáporná data: v jedné komponentě nahromaděná především na nule, v druhé kousek od nuly. Modely těchto komponent hledáme tedy mezi nezápornými rozděleními - viz úloha ODHAD SMĚSI PRO NEZÁPORNÁ DATA.

Program

```
// Estimation of hierarchical mixture
// - upper - SPEED (with three normal components)
// - lower - SPEED DIFFERENCE and CROSS ACCELERATION
//           (economical and sport-like driving is to be detected)
//   absolute values of data -> first quadrant
//   two lower components: 1. HNm, 2. Exp
// ---------------------------------------------------------
exec ScIntro.sce, mode(0)

nd=1500;                      // number of data for estimation
ni=50;                        // virtual length of initialization

load  data0kruhy.dat;         // real data from a car
dat=data0kruhy.data0rig(:,3001:10:$);
dat=dat(:,3601:7240);

// data
v=dat(2,1:nd);                // upper data - velocity
d=abs([diff(v) 0]);           // abs. values of velocity variations
a=20*abs(dat(9,1:nd));        // abs. values of cross-acceleration

// variables
yt=v;                         // data -> varibles (speed)
dt=[d;a]+1e-8;                // data -> varibles (accelerations)

// INITIAL PARAMETERS ============================================
```

```
thC0=[50 70 100];             // speeds in: town, out-town, motorway
thP0=[.2 .5 .3];              // probabilities of going in -"-
nc0=length(thC0);             // number of upper components

nv=2;                         // number of lower variables
nc=[2 2 2];                   // number of lower components

w0=fnorm(ones(1,nc0));        // initial upper weights
w=list();
for i=1:nc0                   // initial lower weights
  w(i)=3+rand(1,nc(i));
  w(i)=w(i)/sum(w(i));
end

thCr=list(); thCe=list();
// parameters of lower components
// - the data are: - d1 long. acceleration (mean value is 2.6
//                 - d2 cross accel. (mean after normaliz. is 3.8
//           d1 d2
thCe(1)=      [1 .2]';        // Exp (1st comp. of 1st mixture)
thCr(1)=%pi/2*[8 4]';        // HNm (2nd komp. of 1st mixture)

thCe(2)=      [1 .2]';         // 2nd mixture
thCr(2)=%pi/2*[8 4]';        //   expectation -> statistics

thCe(3)=      [1 .2]';         // 3rd mixture
thCr(3)=%pi/2*[8 4]';

// INITIALIZATION ================================================
// UPPER mixture
for j=1:nc0
  Ps=[thC0(:,j);1];           // Regression vector
  Cyp.V=Ps*Ps';
  Est0.Cy(j)=Cyp;             // information matrix
  Est0.Cy(j).th=thC0(:,j);    // parameters
  Est0.Cy(j).sd=.1;           // std of noise
end
Est0.ka=1*rand(1,nc0,'u');    // counter
Est0.Cp.V=rand(nc0,nc0,'u')+1;// statistics for pointer estimation
Est0.Cp.th=fnorm(Est0.Cp.V,2);// parameters of pointer model
Est0.ct=1;                    // initial value of the pointer
Est0.nc=nc0;                  // number of upper components
Est0.w=w0;                    // initial weight
Est0.W=zeros(nc0,nc0);
Est0.tht=zeros(nc0,nd);
Est0.wt=zeros(nc0,nd);

// LOWER mixture
for i=1:nc0
```

```
  Esti.i=i;                    // label of lower component
  Esti.ka=ni*ones(1,2);        // counter
  // first component
  Esti.Cy(1).V=ni*thCe(i);     // statistics
  Esti.Cy(1).th=thCe(i);       // initial parameters
  // second component
  Esti.Cy(2).V=ni*thCr(i);        // statistics
  Esti.Cy(2).th=.8*thCr(i);    // initial parameters

  Esti.Cp.V=rand(1,nc(i),'u')+1; // pointer statistics
  Esti.Cp.th=fnorm(Esti.Cp.V);   // pointer parameters

  Esti.nc=nc(i);               // number of lower components
  Esti.w=w(i);                 // initial weights
  Esti.ct=zeros(1,nd);

  Est(i)=Esti;                 // lower component construction
end
//pause
// ESTIMATION loop ============================================
tic();
kk=ceil(nd/10); printf('\n  2 4 6 8 |\n ')
for t=1:nd
if t/kk==fix(t/kk), printf('.'); end

  [Est0,Est]=MEst78(yt(:,t),dt(:,t),Est0,Est);
    [xxx,Est0.ct(1,t)]=max(Est0.w);
    Est0.wt(:,t)=Est0.w';
    Est0.tht(:,t)=[Est0.Cy(1).th,Est0.Cy(2).th,Est0.Cy(3).th]';

  for i=1:nc0
    Est(i).tht(:,t)=[Est(i).Cy(1).th(:);Est(i).Cy(2).th(:)];
    [xxx,Est(i).ct(t)]=max(Est(i).w);
    Est(i).wt(:,t)=Est(i).w';
  end
end
// end of estimation loop =====================================
disp ' ', time=toc()/60; printf(' time = %5.2f min.\n',time)

// RESULTS
d=list();
for i=1:3
  m=find(Est0.ct==i);
  d(i)=dt(:,m);
  for j=1:2
    k=find(Est(i).ct(m)==j);
    Ct.t=d(i)(:,k);
    E(i).C(j)=Ct;              // data in components
  end
```

```
end

// data clusters colored with respect to components
tx=['in town';'outside town';'motorway'];
set(scf(112),'position',[400 150 1000 350]);
for i=1:3
  subplot(1,3,i)
  plot(E(i).C(1).t(1,:),E(i).C(1).t(2,:),'b.','markersize',3)
  plot(E(i).C(2).t(1,:),E(i).C(2).t(2,:),'r.','markersize',3)
  title(tx(i))
  set(gca(),'data_bounds',[0 30 0 20]);
  legend('economic driving','sport driving');
end
```

Procedura

```
function [Eh,Ed]=MEst78(y,d,Eh,Ed)
  // estimation of both upper and lower mixture
  // upper: 3 norm. components
  // lower: 2 comp. - first half-normal
  //                - second exponential

  nc=Eh.nc;
  al=Eh.Cp.th;
  Wh=Eh.w;
  ny=length(d);

  wd=list();
  v=list();
  // PROXIMITIES AND WEIGHTS
  for i=1:nc
    // PROXIMITY UPPER
    [xxx,Gh(i)]=GaussN(y,Eh.Cy(i).th,Eh.Cy(i).sd); // Gauss
    // PROXIMITY LOWER
    [xxx,Gd(1)]=HnormN(d,Ed(i).Cy(1).th)          // (th is "s"; E=.8*s)
    [xxx,Gd(2)]=ExpN(d,Ed(i).Cy(2).th);           // (th is expectation)
    Ld=Gd-max(Gd);
    q(i,:)=fnorm(exp(Ld))';                       // proximity lower

    wd(i)=Ed(i).w;
    be(i,:)=Ed(i).Cp.th;
  end
  Lh=Gh-max(Gh);
  p=fnorm(exp(Lh))';                              // proximity upper

  W=fnorm((Wh'*p).*al);                           // joint pf of c_t and c_t-1
  Wh=sum(W,1);                                    // uppwer weight
  [xxx,ch]=max(Wh);
```

```
  Eh.w=Wh;

  Wd=fnorm((Wh'*ones(1,2)).*(q.*be));             // lower weight
  for i=1:Eh.nc
    Ed(i).w=Wd(i,:);
  end // ---------------------------------------- end of proximity

  // UPDATE OF COMPONENTS AND COMPUTATION OF POINT ESTIMATES
  // UPPER mixture
  Ps=[y';1];                                      // Regression vekcor
  for i=1:Eh.nc
    Eh.Cy(i).V=Eh.Cy(i).V+Wh(i)*Ps*Ps';           // information matrix
    Eh.ka(i)=Eh.ka(i)+Wh(i);                      // counter

    // point estimates of upper parameters
    // for scalar y !!!
    Vy=Eh.Cy(i).V(1,1);                           // Vyp  - y.y
    Vyp=Eh.Cy(i).V($,1);                          // Vyp - psi.y
    Vp=Eh.Cy(i).V($,$);                           // Vp  - psi.psi'
    Eh.Cy(i).th=inv(Vp+1e-8*eye(Vp))*Vyp;         // estimate of reg. coef.
    Eh.Cy(i).tht(:,t)=Eh.Cy(i).th';
    if 0                                          // estimate of covariance
      Eh.Cy(i).cv=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/Eh.ka(i);
    end
  end
  Eh.Cp.V=Eh.Cp.V+W;                              // pointer statistics
  Eh.Cp.th=fnorm(Eh.Cp.V,2);                      // estimate of pt. parameters

  // LOWER mixture
  // update of statistics for hnm
  for i=1:Eh.nc
    Ed(i).Cy(1).V=Ed(i).Cy(1).V+wd(i)(1)*d^2;     // stat. = sum of squares
    Ed(i).ka(1)=Ed(i).ka(1)+wd(i)(1);             // counter
    // point estimates of parameters for hnm
    Ed(i).Cy(1).th=sqrt(Ed(i).Cy(1).V/Ed(i).ka(1));

    // update of statistics for exp
    Ed(i).Cy(2).V=Ed(i).Cy(2).V+wd(i)(2)*d;       // stat. = sum
    Ed(i).ka(2)=Ed(i).ka(2)+wd(i)(2);             // counter
    // point estimates of parameters for exp
    Ed(i).Cy(2).th=Ed(i).Cy(2).V/Ed(i).ka(2);

    Ed(i).Cp.V=Ed(i).Cp.V+wd(i);                  // pointer statistics
    Ed(i).Cp.th=fnorm(Ed(i).Cp.V,2);              // pointer parameters
  end
endfunction
```

Popis programu

Program je dosti složitý. Proto je realizován pomocí procedury MEst78.sci, která obsahuje algoritmus vlastního odhadu komponent jednotlivých směsí.

Hlavní program obstarává

- natažení a úpravu dat (absolutní hodnoty akcelerací),
- inicializaci odhadu pro horní a dolní směsi,
- prezentaci výsledků.

V časové smyčce (ESTIMATION lopp) se volá funkce MEst78(yt(:,t),dt(:,t),Est0,Est), která obstarává on-line běh odhadovacího algoritmu.

Je tvořena třemi částmi

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. $f_c\left(y_t|\hat{\Theta}_{c;t-1}\right)$, kde $c = 1, 2, \cdots, nc$ (pro všechny komponenty).
2. Přepočet statistik, kde data se přidávají s příslušnou váhou.
3. Výpočet bodových odhadů.

Každý z těchto kroků se provádí nejdříve pro horní směs a pak pro jednotlivé dolní směsi. Proximity jsou počítány pomocí funkcí GaussN, HNormN a ExpN, a pomocí nich jsou dále počítány váhy Wh a Wd.

V další části jsou přepočítávány statistiky a počítány bodové odhady parametrů pro jednotlivé směsi: horní s normálním rozdělením a dolní s polovičním normálním a exponenciálním rozdělením.