Klasifikace s regresním modelem

- simulovaná data
- klasifikace - třídění dat do tříd
- logistický model a jeho ML odhad

V programu se provádí odhad modelu logistické regrese (fáze učení) a pro odhadnutý model, tj. model s pevnými parametry, se provádí třídění dat (fáze testování). Data se zařadí do třídy, která je indikována hodnotou odhadnutého výstupu (blíže viz poznámka).

Simulace se provádí tak, ze se generují náhodné vektory x_1 a x_2. Vytvoří se jejich lineární kombinace $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ (ϵ je šum) a pro výsledek větší než 0.5 se přiřadí $y = 1$ jinak $y = 0$.

Model v úloze učení testování má standardní tvar logistické regrese

$$z_t = x_t b$$

$$P\left(y_t = 1 | x_t\right) = \frac{\exp\left\{z_t\right\}}{1 + \exp\left\{z_t\right\}}$$

$$\hat{y}_t = \begin{cases} 1 & \text{pro } P\left(y_t = 1 | x_t\right) > 0.5 \\ 0 & \text{pro } P\left(y_t = 1 | x_t\right) < 0.5 \end{cases}$$

Předpoklady: Dvouhodnotové $y \in \{0, 1\}$

Sci značení: yL, xL - data pro učení; xT - data pro testování; c - odhad výstupu (třídy); yT - správná třída ze simulace.

Úloha: Odhad modelu logistické regrese, odhad třídy dat pro měřené datové záznamy (klasifikace).

Poznámka

V logistické regresi se používají veličiny y - výstup a x - regresní vektor. V klasifikaci mají jinou interpretaci: x jsou datové záznamy, které chceme třídit, y je označení třídy, do které data patří.

Doporučené experimenty

1. Zkuste měnit velikost datového vzorku pro učení (pomocí parametru nL). Sledujte vliv na kvalitu klasifikace. Ta se posuzuje jako počet špatných klasifikací vzhledem k počtu provedených klasifikací.
2. Simulace je v programu provedena heuristicky. Nejdříve se generují hodnoty nezávislých proměnných x. Potom se provede jejich lineární kombinace a výstup y se určí jako 1 pro její kladné hodnoty, v opačném případě je 0. Zkuste navrhnout svou vlastní simulaci.
3. Zkuste provést simulaci generovanou logistickým modelem, pro který si zadáte své vlastní parametry. Porovnejte simulované a odhadnuté parametry.

Program

```
// Classification with logistic regression
[u,t,n]=file();                          // find working directory
chdir(dirname(n(1)));                    // set working directory
clear("u","t","n")                       // clear auxiliary data
exec("ScIntro.sce",-1),mode(0)           // intro to sesion

nL=100;                        // number of data for learning
nT=100;                        // number of data for testing
sd=1;                          // magnitude of disturbance

x1=rand(1,nL+nT,'norm');       // 1. regression vector
x2=rand(1,nL+nT,'unif')<.3;    // 2. regression vector
x=[x1;x2]';                    // full regression vector
sL=1:nL;                       // inteval for learning
sT=1:nT;                       // interval for testing
yy=2*x1-3*x2+1+sd*rand(1,nL+nT,'norm');
y=double(yy>0);                // output (true classes)
yL=y(sL);
xL=x(sL,:);

// Learning
// py     probabilities of y=0 and y=1
// yp     prediction of y  yp=py(:,2)
// yr     point prediction of y  yr=round(yp)
// b      model parameters
[py,yp,yr,b]=lrEst([],yL,xL); // log. reg. estimation

// Testing
ct=zeros(sT);
yT=y(1,nL+sT);                 // true classes
xT=[ones(nT,1) x(nL+sT,:)];    // data for testing
for t=sT
  z=xT(t,:)*b;                 // regression
  yp=exp(z)/(1+exp(z));        // logistic function
  ct(t)=round(yp);             // class point estimates
end

// RESULTS
s=1:nT;
set(scf(1),'position',[600 50 600 600])
plot(s,yT(s),'bo',s,ct(s),'r.')
set(gca(),"data_bounds",[min(s)-.1 max(s)+.1 -.1 1.1])
title('Logistic regression')
legend('output','point estimate');

disp('Logistic regression:')
printf(' Wrong %d from %d\n',sum(yT~=ct),nT)
```