粒子物理与核物理实验中的数据分析

陈少敏
清华大学

第八讲：最大似然法(II)

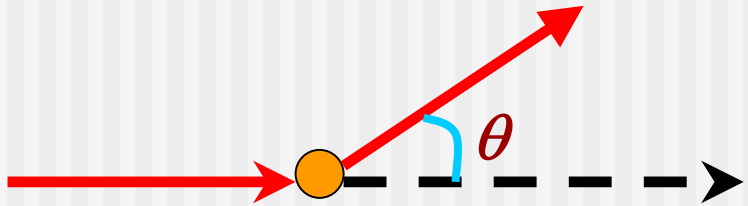
本讲要点

- 双参数情况下的最大似然法举例
- 利用MINUIT数值求解最小值
- 推广的最大似然法
- 利用最大似然法处理分区数据
- 检验最大似然法的拟合优度
- 与贝叶斯参数估计之间的关系

例子:最大似然法处理双参数

考虑散射角分布 $x=\cos\theta$

$$f(x;\alpha,\beta)=\frac{1+\alpha x+\beta x^2}{2+2\beta/3}$$



在实际情况下，探测器也许不可能做全空间探测，即 $x_{\min}<x<x_{\max}$ 。

这个时候需要考虑效率影响，并对函数进行归一化处理，例如

$$\varepsilon_{效率}(x)=\begin{cases}1 & x_{\min}<x<x_{\max}\\ 0 & 否则\end{cases}$$

$$\Rightarrow \int_{-1}^{1} f(x;\alpha,\beta)\cdot\varepsilon(x)dx=1$$

例如：对于参数为α=0.5，β=0.5，探测器有效范围 $x_{\min}$=-0.95, $x_{\max}$= 0.95, 产生 n=2000个蒙特卡罗事例，研究用最大似然法处理双参数的问题。

双参数问题(续)

由联合概率密度得到样本的似然函数

$$L(\vec{x};\alpha,\beta)=\prod_{i=1}^{2000} f(x_i;\alpha,\beta)$$

用数值解找到 $\log L$ 的最大值

$$\hat{\alpha}=0.50\pm 0.05$$

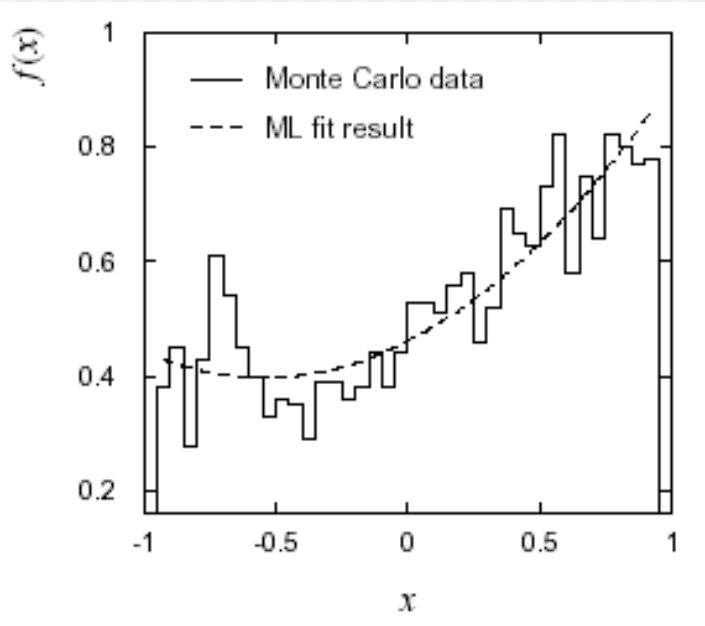
$$\hat{\beta}=0.51\pm 0.11$$

$$\widehat{\mathrm{cov}}\left[\hat{\alpha},\hat{\beta}\right]=0.0024$$

$$\hat{\rho}=r=0.42$$

“误差”：

$\hat{\sigma}_{\hat{\alpha}},\hat{\sigma}_{\hat{\beta}}$



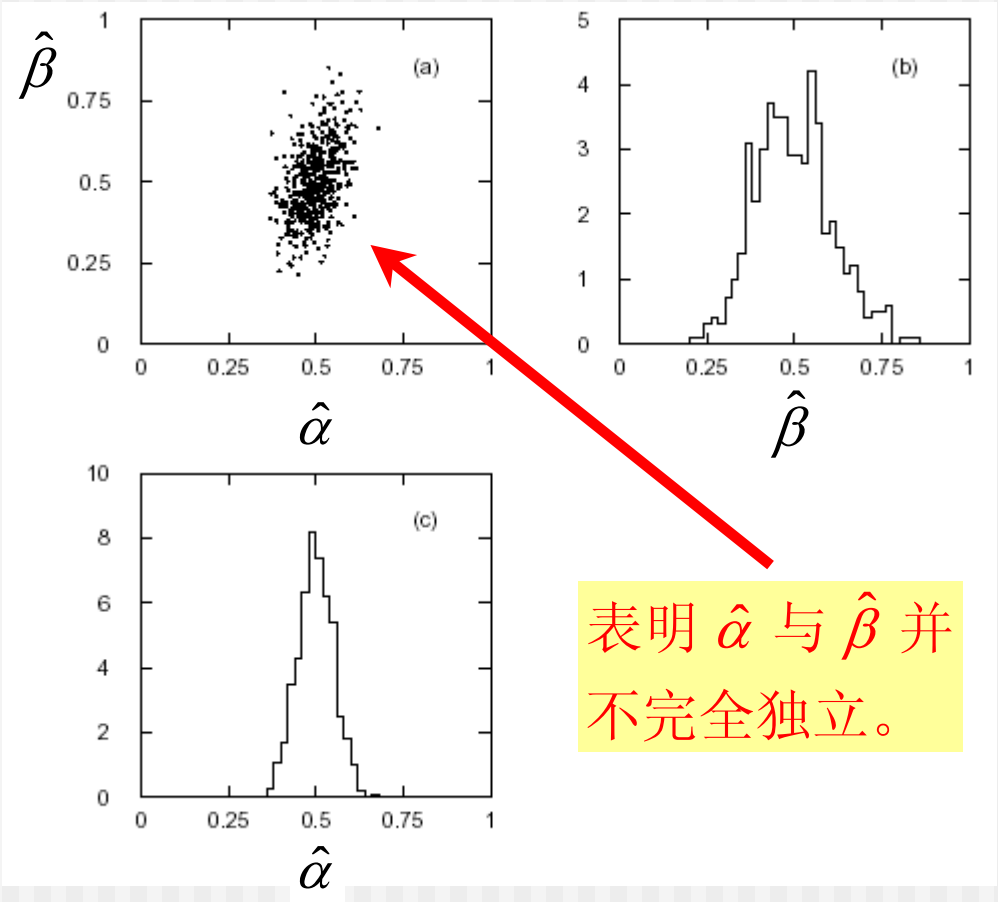
注意：拟合中采用的是点拟合，与直方图的区间大小划分无关。

协方差 $$(\widehat{V^{-1}})_{ij}=-\left.\frac{\partial^2 \log L}{\partial\theta_i\partial\theta_j}\right|_{\vec{\theta}=\hat{\vec{\theta}}}$$

(MINUIT HESSE)

双参数拟合：蒙特卡罗研究

做500次模拟实验，重复最大似然拟合，每次都是模拟 n=2000 个事例：



表明 $\hat{\alpha}$ 与 $\hat{\beta}$ 并不完全独立。

$\overline{\hat{\alpha}} = 0.499$ $\overline{\hat{\beta}} = 0.498$

$s_{\hat{\alpha}} = 0.051$ $s_{\hat{\beta}} = 0.111$

$\overline{\mathrm{cov}[\hat{\alpha},\hat{\beta}]} = 0.0024$

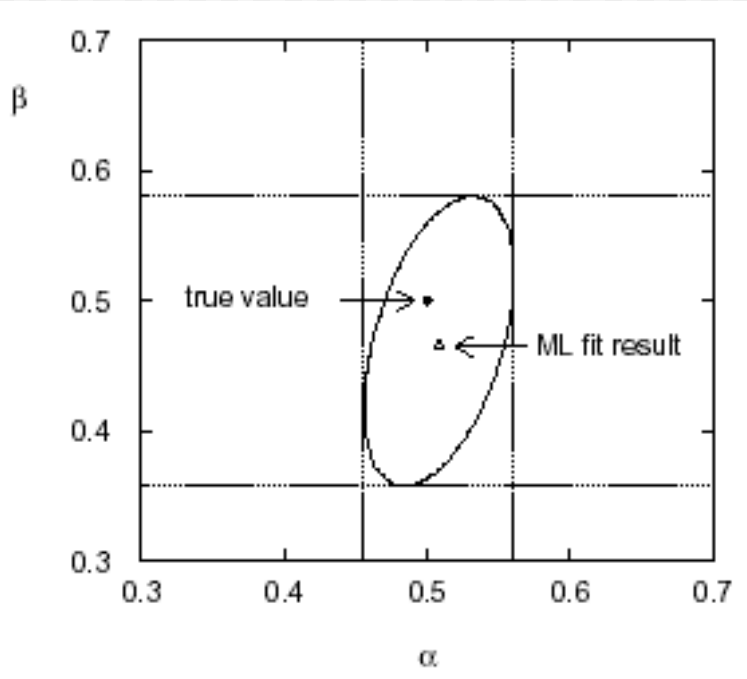
$\hat{\rho} = r = 0.42$

$\hat{\alpha},\hat{\beta}$ 正相关

边缘概率密度函数近似为高斯分布。

双参数拟合:log L 等高线

在前面蒙特卡罗样本中，其中的一次拟合结果在α，β 平面表示为



对于大的样本容量 n，$\log L$ 具有下列形式：

$$\log L(\alpha,\beta) = \log L_{\max} - \frac{1}{2(1-\rho^2)}\left[\left(\frac{\alpha-\hat{\alpha}}{\sigma_{\hat{\alpha}}}\right)^2 + \left(\frac{\beta-\hat{\beta}}{\sigma_{\hat{\beta}}}\right)^2 - 2\rho\left(\frac{\alpha-\hat{\alpha}}{\sigma_{\hat{\alpha}}}\right)\left(\frac{\beta-\hat{\beta}}{\sigma_{\hat{\beta}}}\right)\right]$$

双参数拟合:log L 等高线(续)

等高线 $\log L(\alpha,\beta)=\log L_{\max}-½$ 是根据下式定义得到的

$$\frac{1}{1-\rho^2}\left[\left(\frac{\alpha-\hat{\alpha}}{\sigma_{\hat{\alpha}}}\right)^2+\left(\frac{\beta-\hat{\beta}}{\sigma_{\hat{\beta}}}\right)^2-2\rho\left(\frac{\alpha-\hat{\alpha}}{\sigma_{\hat{\alpha}}}\right)\left(\frac{\beta-\hat{\beta}}{\sigma_{\hat{\beta}}}\right)\right]=1$$

其中，等高线的切线给出了标准偏差

$$\sigma_{\hat{\alpha}},\sigma_{\hat{\beta}}$$

椭圆的倾角与相关系数有关

$$\tan 2\phi=\frac{2\rho\sigma_{\hat{\alpha}}\sigma_{\hat{\beta}}}{\sigma_{\hat{\alpha}}^2-\sigma_{\hat{\beta}}^2}$$

注意：参数之间相关的效应体现在对估计量的误差或方差方面的影响(或偏大或偏小)。

一个实用的求函数最小值程序

在核与粒子物理研究中，大家普遍使用的求函数最小值程序是

MINUIT 软件包

http://hep.tsinghua.edu.cn/~chensm/lectures/minuit.pdf

求 $\log L$ 最大值等效于求 $-\log L$ 的最小值。

它可以对目标函数为似然函数，最小二乘函数或用户自定义函数求极值。它提供了几种最小化过程和相应的误差分析。原初版本用Fortran语言。写于25年前，用于当时西欧核子中心(CERN)的实验数据分析。现也被应用于粒子物理以外的领域。目前，在核与粒子物理研究中，有三种使用方式

- 在MINUIT框架下单独使用(适于data-driven模式)；
- 在物理分析工作站(PAW)环境下互动调用MINUIT软件包(基于Fortran)；
- 在PAW升级软件包ROOT环境下互动调用MINUIT软件包(基于C++)。

MINUIT数值求解极小值(1)

实际应用中，通常采用$-\log L(\vec{\theta})$的形式来求最小值。例如，对前面的例子

```
      program MINUIT_FIT
      implicit NONE
      integer npar
      parameter (npar=2)
      character*10 chnam(npar)
      integer ierr, ird, isav, istat, ivarbl, iwr
      integer npari, nparx
      double precision arglis(10), bnd1, bnd2, deriv(npar), dpar(npar)
      double precision fmin, fedm, errdef, covmat(npar,npar), log_l
      external FCN
      double precision par(npar)
c Initialize MINUIT, set print level to -1
      ird = 5 ! unit number for input to Minuit (keyboard)
      iwr = 6 ! unit number for output from Minuit (screen)
      isav= 7 ! unit number for use of the SAVE command
      call MNINIT(ird,iwr,isav)
      arglis(1)=-1.d0
      call MNEXCM(FCN,'SET PRIN',arglis,1,ierr,0)
c Define parameters alpha and beta, give initial values and step sizes
      call MNPARM(1,'alpha',0.5d0,0.1d0,0.d0,0.d0,ierr)
      call MNPARM(2,'beta ',0.5d0,0.1d0,0.d0,0.d0,ierr)
c Get input data by calling FCN with iflag=1
      arglis(1)=1.d0
      call MNEXCM(FCN,'CALL',arglis,1,ierr,0)
c Minimize using SIMPLEX and MIGRAD, get covariance
c matrix with HESSE
      call MNEXCM(FCN,'SIMPLEX',arglis,0,ierr,0)
      call MNEXCM(FCN,'MIGRAD',arglis,0,ierr,0)
      call MNEXCM(FCN,'HESSE',arglis,0,ierr,0)
c Get result of fit (for least squares, fmin is chi2)
      call MNSTAT(fmin,fedm,errdef,npari,nparx,istat)
      call MNPOUT(1,chnam(1),par(1),dpar(1),bnd1,bnd2,ivarbl)
      call MNPOUT(2,chnam(2),par(2),dpar(2),bnd1,bnd2,ivarbl)
      call MNEMAT(covmat,npar)
      log_l=-0.5*fmin
      write(6,*)'Fit results:'
      write(6,*)
      write(6,*)'alpha         =',par(1),'+/-',dpar(1)
      write(6,*)'beta          =',par(2),'+/-',dpar(2)
      write(6,*)'cov[alpha,beta]=',covmat(1,2)
      write(6,*)'rho[alpha,beta]=',covmat(1,2)/(dpar(1)*dpar(2))
      write(6,*)'log_l         =',log_l
      stop
      end
```

编译：**f77 minuit_fit.f -o minuit_fit `cernlib` <return>**

运行：**minuit_fit <return>**

MINUIT数值求解极小值(2)

用户应提供似然函数 FCN

```
      subroutine FCN(npar,grad,chi2,par,iflag,futil)
c Input:  integer          npar       number of parameters to fit
c         double precision  par(npar)  parameter vector
c         integer          iflag      select what to do
c         double precision  futil      optional external function
c Output: double precision  grad(npar) gradient vector (not filled)
c         double precision  chi2       function to be minimized
      implicit NONE
      integer npar,iflag
      double precision futil,chi2,par(*),grad(*)
      integer n_max
      parameter (n_max=10000)
      integer i,n
      double precision alpha,beta,f,log_l,x(n_max),angle_cut,f_nor
c Begin
      n=2000
      angle_cut=0.95
      if(iflag.eq.1)then ! get n, array x
       call GET_INPUT_DATA(x,n,n_max,angle_cut)
      end if
```

注意：概率密度函数需要在有效测量范围进行归一化处理。如果计算有困难，可以将归一化因子作为参数来拟合。这种折衷方法虽简便，但会给其它待定参数的确定带来误差。

```
c Calculate log-likelihood
      alpha = par(1)
      beta  = par(2)
      log_l = 0.
c Normalization factor for [-angle_cut,angle_cut]
      f_nor = 2*angle_cut+2*beta/3.*angle_cut**3
      do i=1,n
        f=(1.+alpha*x(i)+beta*x(i)**2)/f_nor
        log_l=log_l+DLOG(f)
      end do
      chi2=-2.*log_l  ! 2 gets errors right
      return
      end
```

MINUIT数值求解极小值(3)

对于采用蒙特卡罗研究参数估计值问题，可用下面的舍选法例子对任意概率密度函数进行抽样，得到模拟的数据样本。

```
      subroutine get_input_data(x,n,n_max,angle_cut)
      integer n,n_max,ntot
      double precision x(n_max),angle_cut
      real rvec(1),alpha,beta,r,fmax,z,u
      alpha=0.5
      beta =0.5
      fmax=-999.
      do i=1,100
        call ranmar(rvec,1)
        r=rvec(1)*2.0-1.0  ! from (0,1) to (-1,1)
        f=(1.+alpha*r+beta*r**2)/(2.+2.*beta/3.)
        if(fmax.lt.f)fmax=f
      end do
      fmax=1.05*fmax
      ntot=0
   10 call ranmar(rvec,1)
      r=rvec(1)*2.0-1.0  ! from (0,1) to (-1,1)
      z=(1.+alpha*r+beta*r**2)/(2.+2.*beta/3.)
      if(z.gt.fmax)then
        fmax=z
        write(6,*)'z greater than fmax'
      end if
      call ranmar(rvec,1)
      u=rvec(1)*fmax
      if(u.le.z)then
        if(abs(r).lt.angle_cut)then
          ntot=ntot+1
          x(ntot)=r
          if(ntot.ge.n)go to 20
        end if
      end if
      go to 10
   20 continue
      return
      end
```

MINUIT数值求解极小值(4)

做500次模拟实验，重复最大似然拟合，每次都是模拟 n=2000 个事例：

对前面的程序添加下列指令

```
program MINUIT_FIT
…
double precision par(npar)
real h(80000)
common/pawc/h
integer i
call hlimit(80000)
call hbook2(200,'alpha vs beta',100,0.,1.,100,0.,1.,0.)
do i=1,500
c Initialize MINUIT, set print level to –1
…
  call hfill(200,real(par(1)),real(par(2)),1.0)
end do
call hrput(0,'minuit_fit.hbook','N')
…
```

```
mess $shell('f77 minuit_fit.f -o minuit_fit `cernlib`')
mess $shell('minuit_fit')
fun2 300 1./(1-0.42**2)*((x-0.5)**2/0.051**2+_
(y-0.5)**2/0.111**2-2*0.42*(x-0.5)/0.051*_
(y-0.5)/0.111)-1. 100 0. 1. 100 0. 1.
h/fil 1 minuit_fit.hbook
zone 2 2; slix 200 1; sliy 200 1; h/proj 200
h/pl 200.sliy.1; h/pl 200
Ve/cr tmp(50); contour 300 1 1 tmp
h/pl 200.slix.1
```

“minuit_fit.kumac”

可以从
/home/chensm/examples/paw
拷贝
minuit_fit.f 和 minuit_fit.kumac
运行 PAW>exec minuit_fit

MINUIT数值求解极小值(5)

在 ROOT 环境下的运行程序

```
const int npoints=2000;
Double_t x[npoints];
Double_t angle_cut=0.95;
void minuit_fit()
{
  // Get data points
   get_input_data();
   // Prepare for fit
   const int npar=2;
   TMinuit *gMinuit = new TMinuit(npar);
   gMinuit->SetFCN(fcn);
   Double_t arglist[10];
   Int_t ierflg = 0;
    arglist[0] = 1;
   gMinuit->mnexcm("SET ERR", arglist ,1,ierflg);
// Set starting values and step sizes for parameters
   static Double_t vstart[npar] = {0.5, 0.5 };
   static Double_t step[npar] = {0.1 , 0.1 };
   gMinuit->mnparm(0, "alpha", vstart[0], step[0], 0,0,ierflg);
   gMinuit->mnparm(1, "beta", vstart[1], step[1], 0,0,ierflg);
// Now ready for minimization step
   arglist[0] = 500;
   arglist[1] = 1.;
   gMinuit->mnexcm("SIMPLEX", arglist ,0,ierflg);
   gMinuit->mnexcm("MIGRAD", arglist ,0,ierflg);
   gMinuit->mnexcm("HESSE", arglist ,0,ierflg);
// Print results
   Double_t fmin,fedm,errdef,covmat[npar][npar];
   Double_t alpha,alpha_err,beta,beta_err;
   Int_t nvpar,nparx,icstat;
   gMinuit->mnstat(fmin,fedm,errdef,nvpar,nparx,icstat);
   gMinuit->GetParameter(0, alpha, alpha_err );
   gMinuit->GetParameter(1, beta, beta_err );
   gMinuit->mnemat(&covmat[0][0],npar);
   Double_t rho=covmat[0][1]/(alpha_err*beta_err);
   cout <<"alpha = "<<alpha<<" +/- "<<alpha_err<<endl;
   cout <<"beta   = "<<beta <<" +/- "<<beta_err<<endl;
   cout <<"cov[alpha][beta]= "<<covmat[0][1]<<endl;
   cout <<"rho[alpha][beta]= "<<rho<<endl;
   cout <<"log_l          = "<<-0.5*fmin<<endl;
}
```

root>.x minuit_fit.C

MINUIT数值求解极小值(6)

似然函数

```
void fcn(Int_t &npar, Double_t *gin, Double_t &chi2, Double_t *par, Int_t iflag)
{
   // Calculate log-likelihood
   Double_t log_l = 0;
   Double_t alpha,beta,f_nor,f;
   alpha = par[0];
   beta  = par[1];
   f_nor = 2*angle_cut+2*beta/3.*angle_cut**3;
   for (Int_t i=0;i<npoints; i++) {
     f=(1.+alpha*x[i]+beta*x[i]**2)/f_nor;
     log_l += TMath::Log(f);
   }
   // 2 gets errors right
   chi2=-2.*log_l;
}
```

MINUIT数值求解极小值(7)

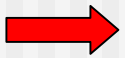
产生按用户定义概率密度函数的分布数据

```
void get_input_data()
{
// Generate data points
   Double_t alpha,beta,r,f,fmax;
   alpha=0.5;
   beta =0.5;
   fmax=-999.;
   TH1F *h1 = new TH1F("h1","x",100,-1,1);
   for (Int_t i=0;i<100; i++) {
     r = gRandom->Rndm();
     r = 2.*r-1.;
     f = (1.+alpha*r+beta*r**2)/(2.+2.*beta/3.);
     if(fmax<f)fmax=f;
   }
   fmax=1.05*fmax;
```

```
Int_t ntot=0;
   Double_t z,u;
   Float_t x_val;
   while(ntot<npoints){
     r = gRandom->Rndm();
     r = r*2.-1;
     z = (1.+alpha*r+beta*r**2)/(2.+2.*beta/3.);
     if(z>fmax){
       fmax=z;
       cout<<"z greater than fmax"<<endl;
     }
     u = gRandom->Rndm();
     u = u*fmax;
     if(u<=z){
       if(TMath::Abs(r)<angle_cut){
         x[ntot]=r;
         x_val = r;
         h1->Fill(x_val);
         ntot++;
       }}}}
```

推广的最大似然法

到目前为止，我们只考虑了固定样本大小 n 的情形，有时候，n 被看作泊松分布的随机变量，平均值为ν。



实验结果定义：$n, x_1, \ldots, x_n$

则推广的似然函数为

$$L(\nu, \vec{\theta}) = \frac{\nu^n}{n!} e^{-\nu} \prod_{i=1}^{n} f(x_i; \vec{\theta})$$

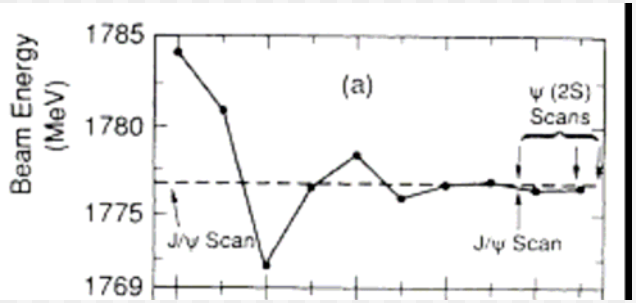
假设理论给出 $\nu = \nu(\vec{\theta})$，却掉与θ无关的项，有

$$\log L(\vec{\theta}) = n \log \nu(\vec{\theta}) - \nu(\vec{\theta}) + \sum_{i=1}^{n} \log f(x; \vec{\theta})$$

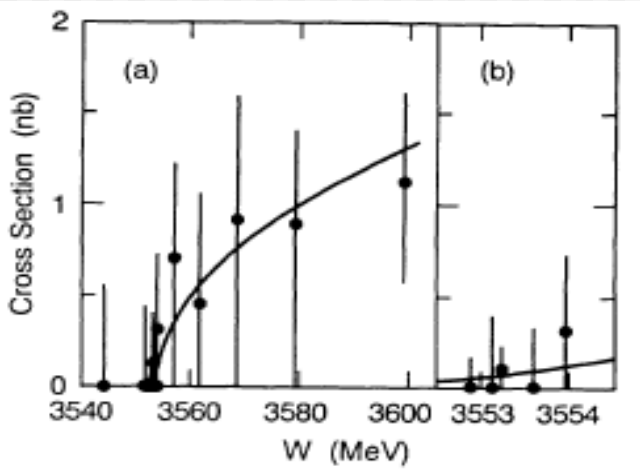
$$= -\nu(\vec{\theta}) + \sum_{i=1}^{n} \log(\nu(\vec{\theta}) f(x_i; \vec{\theta}))$$

PRL 69, 3021 (1992)

举例：τ 轻子质量测量



能量点 $i\rightarrow$



反应过程：$e^+ + e^- \rightarrow \tau^+ + \tau^-$

已知：e^+与e^-能量相等，动量方向相反；过程截面σ已知；e^+与e^-的束流亮度Φ已知。在一定时间内，预期的事例数为$\nu = \sigma(W)\int \Phi dt$。

实验原理：通过改变e^+e^-质心能量W，确定产生$\tau^+\tau^-$的能量阈来测量τ 轻子的质量。

第i个能量点的实验结果：$n_i, W_1 = ... = W_n$

$$L(\nu_i, m_\tau) = \frac{\nu_i^{n_i}}{n_i!} e^{-\nu_i} \prod_{j=1}^{n_i} f(W_j; m_\tau)$$

探测效率 $f(W_j; m_\tau)$对同一能量点为常数。Data-Driven寻找方式给出

$(n_i, W_i) \Rightarrow$ 阈值$W_{i+1}=2\hat{m}_\tau \Rightarrow (n_{i+1}, W_{i+1}) \Rightarrow \ ... \ \Rightarrow \hat{m}_\tau$

推广的最大似然法：$\nu, \vec{\theta}$ 独立

假设$\nu, \vec{\theta}$是在函数上相互独立的，

$$L(\nu, \vec{\theta}) = \frac{\nu^n}{n!} e^{-\nu} \prod_{i=1}^{n} f(x_i; \vec{\theta})$$

$$\frac{\partial L}{\partial \nu} = \left(\frac{n}{\nu} - 1 \right) \frac{\nu^n}{n!} e^{-\nu} \prod_{i=1}^{n} f(x_i; \vec{\theta}) = 0 \qquad \longrightarrow \hat{\nu} = n$$

$$\frac{\partial L}{\partial \theta_i} = 0 \qquad \longrightarrow \text{与普通最大似然法确定} \hat{\theta}_i \text{的要求一样}$$

这种情况对处理 $f(x; \vec{\theta})$ 是已知分量叠加的特例是有用的。

可以将其分解成单独求ν与θ估计值问题.

推广的最大似然法：$\nu, \vec{\theta}$ 独立(续)

如果联合概率密度函数可以表示为

$$f(x;\vec{\theta}) = \sum_{i=1}^{m} \theta_i f_i(x)$$

根据概率的定义，可以知道并非所有的 θ_i 独立，因此有

$$\sum_{i=1}^{m} \theta_i = 1 \quad \Rightarrow \quad \theta_m f_m(x) = \left(1 - \sum_{i=1}^{m-1} \theta_i\right) f_m(x)$$

在推广的最大似然法中

$$\log L(\nu, \vec{\theta}) = -\nu + \sum_{i=1}^{n} \log\left(\sum_{j=1}^{m} \nu\theta_j f_j(x_i)\right)$$

定义 $\mu_i = \nu\theta_i$

$$\log L(\vec{\mu}) = -\sum_{j=1}^{m} \mu_j + \sum_{i=1}^{n} \log\left(\sum_{j=1}^{m} \mu_j f_j(x_i)\right)$$

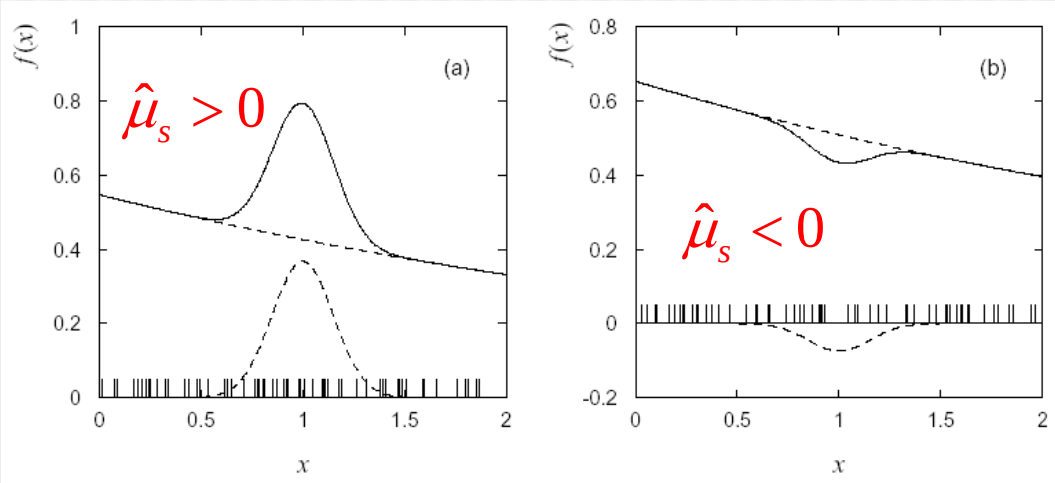
$\mu_1,\ldots,\mu_m$=类型 i 的事例数期待值。

推广的最大似然法举例

假设有两类事例：信号(s)与本底(b)

$$f(x)=\frac{\mu_s}{\mu_s+\mu_b}f_s(x)+\frac{\mu_b}{\mu_s+\mu_b}f_b(x)$$

假设 $f_s(x)$ 与 $f_b(x)$ 已知，需要估计 μ_s 与 μ_b。会出现两种情况



问题：如果出现负值，应该如何报告结果？

应该报告负的 $\hat{\mu}_s$ (非物理结果!)，这种情况下含偏置的真值估计量为：$\hat{\mu}_s^{physical}=\max(0,\hat{\mu}_s)$

最大似然法处理分区数据

通常数据 $\vec{x}$ 在划分为N个区间直方图中的频数为

$$\vec{n} = (n_1, \ldots, n_N), \quad n_{tot} = \sum_{i=1}^{N} n_i$$

通常称为“对直方图拟合”。

在某种假设下，有期待值

$$\vec{\nu} = (\nu_1, \ldots, \nu_N), \nu_{tot} = \sum_{i=1}^{N} \nu_i(\vec{\theta})$$

$$\nu_i(\vec{\theta}) = \nu_{tot} \int_{x_i^{\min}}^{x_i^{\max}} f(x;\vec{\theta})dx$$

如果样本的联合概率密度函数(n_{tot} 为常数)为

$$f_{sample}(\vec{n};\vec{\nu}) = \frac{n_{tot}!}{n_1!\ldots n_N!}\left(\frac{\nu_1}{n_{tot}}\right)^{n_1} \ldots \left(\frac{\nu_N}{n_{tot}}\right)^{n_N}$$

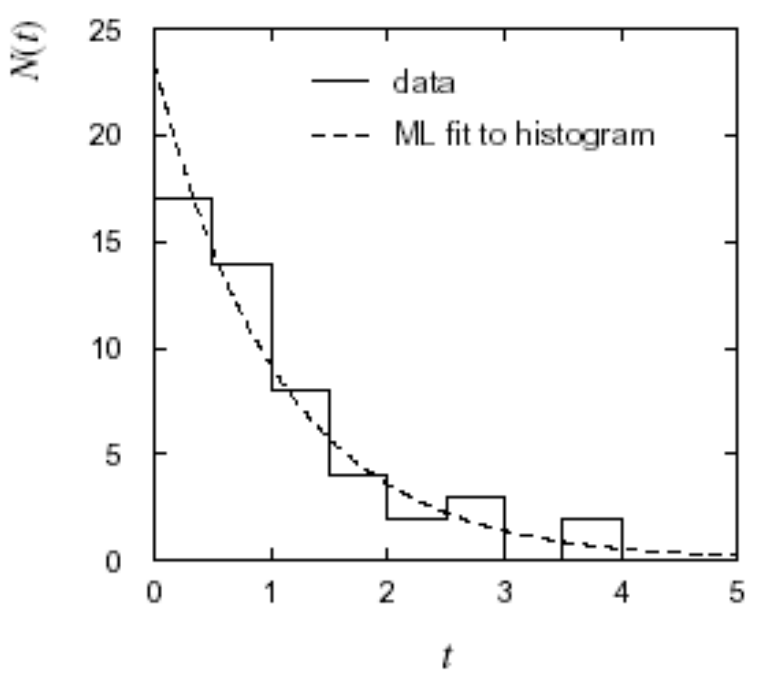
$$\log L(\vec{\theta}) = \sum_{i=1}^{N} n_i \log \nu_i(\vec{\theta})$$

例子：指数分布

指数分布例子

直方图拟合：$\hat{\tau} = 1.07 \pm 0.17$

点估计：$\hat{\tau} = 1.06 \pm 0.15$



两者结果吻合。点估计结果误差较小。

当区间宽度为零时，

直方图拟合 ➡ 点估计

如果 n_i 是泊松随机变量，有推广的对数似然函数

$$\log L(\nu_{tot}, \vec{\theta}) = -\nu_{tot} + \sum_{i=1}^{N} n_i \log \nu_i(\nu_{tot}, \vec{\theta})$$

分区处理数据的问题

分区处理数据，减少了计算量，但当区间宽度过大时，有时可能会造成因部分信息丢失，影响对参数的估计。例如对于前例

$$f(x;\alpha,\beta)=\frac{1+\alpha x+\beta x^2}{2+2\beta/3}$$

由于 β 对函数变化影响较小，过宽的分区，对 β 的估计影响较大。

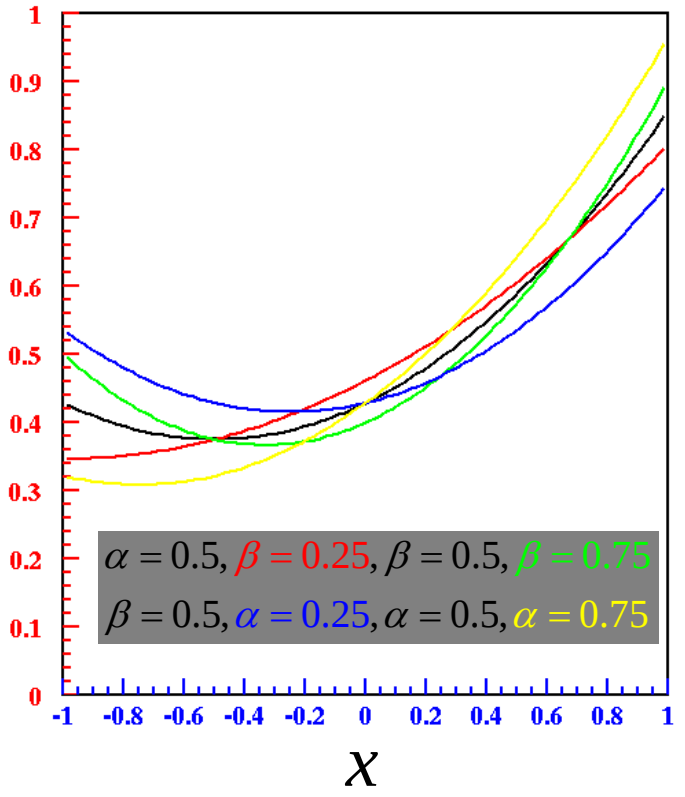
例如，在 $x\in[-1,1]$ 范围内分50个区间

$$\hat{\alpha}=0.47\pm0.05,\quad \hat{\beta}=0.39\pm0.11$$

在 $x\in[-1,1]$ 范围内分200个区间

$$\hat{\alpha}=0.50\pm0.05,\quad \hat{\beta}=0.50\pm0.11$$

因此，对直方图拟合，一定要确认区间的大小对结果无明显影响。注意：区间无穷小时，与点估计结果一致。



检验最大似然法的拟合优度

对非分区(点估计)情况，检验很困难。可以尝试检验统计量 $t = \log L_{\max}$

原则上可行，但检验统计量的分布 $g(t)$ 通常未知

需要蒙特卡罗计算P-值

对分区情况(或者对拟合非分区数据后所做的直方图情况)，

$$\hat{\nu}_i = n_{tot} \int_{x_i^{\min}}^{x_i^{\max}} f(x;\vec{\hat{\theta}})dx \quad (\text{假设有}m\text{个拟合参数})$$

可以考虑研究比值的分布情况

$$\lambda = \frac{L(\vec{n}\,|\,\vec{\nu})}{L(\vec{n}\,|\,\vec{n})} = \frac{f_{joint}(\vec{n};\vec{\nu})}{f_{joint}(\vec{n};\vec{n})}$$

检验最大似然法的拟合优度(续)

例如对于满足多项式或泊松分布的随机变量 n_i，可用检验统计量

$$\chi^2_M = -2\log\lambda_M = 2\sum_{i=1}^{N} n_i \log\frac{n_i}{\hat{\nu}_i}$$

$$\chi^2_P = -2\log\lambda_P = 2\sum_{i=1}^{N}\left(n_i \log\frac{n_i}{\hat{\nu}_i} + \hat{\nu}_i - n_i\right)$$

对于大统计量样本，两者都服从最小二乘的pdf，自由度=N - m

或利用皮尔逊的χ^2检验，→ 用 $\hat{\nu}_i = \nu_i(\hat{\vec{\theta}})$ 代替 ν_i

→

$$\chi^2 = \sum_{i=1}^{N}\frac{(n_i - \hat{\nu}_i)^2}{\hat{\nu}_i}$$

泊松分布：自由度=N - m

$$\chi^2 = \sum_{i=1}^{N}\frac{(n_i - \hat{p}_i n_{tot})^2}{\hat{p}_i n_{tot}}$$

多项式分布：自由度=N - m - 1

似然法与贝叶斯估计的关系

在贝叶斯统计中，θ 与 $\vec{x}$ 都是随机变量

$$L(\theta) = L(\vec{x} \mid \theta) = f_{joint}(\vec{x} \mid \theta)$$

是 $\vec{x}$ 在给定 θ 的条件概率密度函数。

贝叶斯方法

- 对假设 (θ) 采用主观概率来描述；
- 实验前，对过程的了解由 $\pi(\theta)$ 归纳(先验概率密度函数)；
- 利用贝叶斯定理，根据数据来改进先验概率密度函数：

验后概率：$p(\theta \mid \vec{x}) = \dfrac{L(\vec{x} \mid \theta)\pi(\theta)}{\int L(\vec{x} \mid \theta')\pi(\theta')d\theta'}$

似然法与贝叶斯估计的关系(续)

对很纯粹的贝叶斯论者

$p(\theta|\vec{x})$包含了对 θ 的所有知识。

对很实际的贝叶斯论者

$p(\theta|\vec{x})$是一个复杂的函数。 ➡ 利用估计量 $\hat{\theta}_{\text{Bayes}}$ 来归纳出对θ 的认识。

实际应用中，会碰到如何得到 $\pi(\theta)$ 的问题?

无金科玉律(主观的!)，通常假定它在全部取值区域上均匀分布。

➡ $\hat{\theta}_{\text{Bayes}} = \hat{\theta}_{\text{ML}}$

注意：均匀假设会遇到诸如 $\lambda=1/\theta$ 时，$\pi(\lambda)$ 不是均匀分布之类的问题。

非贝叶斯方法目前在统计学中占主要地位。但是，如果被测定参数是随机变量，而且它的验前分布已知，就应该用贝叶斯方法。

小结

1. **最大似然法处理双参数情况**

   从RCF边界，对数似然函数等高线或MC确定方差。

2. **用MINUT找最小值的数值解**

   如何在PAW或ROOT环境下找-log$L$最小值，给出参数估计

3. **推广的最大似然法**

   当样本容量被当作泊松变量时，如何给出参数估计

4. **最大似然法处理分区数据**

   将直方图按多维分布处理，子区间服从泊松分布。部分信息会丢失。

5. **检验最大似然法的拟合优度**

   很困难，除非将数据分区，进行点估计后的拟合优度检验。

6. **与贝叶斯参数估计的关系**

   如果验前分布为常数，则与似然法估计量一致。