蒙特卡洛算法：积分

计算物理b
高阳

一维积分问题

- 定积分

$$I=\int_a^b dy\, f_1(y)\,,\qquad 0\le L\le f_1(y)\le M$$

- “归一化”变换

$$f(y)=\frac{1}{M-L}[f_1(y)-L],\quad 0\le f(y)\le 1$$

$$x=\frac{y-a}{b-a},\quad I=(b-a)(M-L)\int_0^1 dx\, f(x)+L(b-a)$$

- 概率算法：x是（0,1）内的均匀分布，则

$$I_0=E\big(f(x)\big)$$

- 若x按照某个概率g(x)分布，则

$$f^*=\frac{f(x)}{g(x)},\qquad I_0=E\big(f^*(x)\big)$$

离散化

- 方差

$$V(f^*) = \int_0^1 (f^\star - I_0)^2 g(x) dx$$

- 离散撒点之后

$$I_0 \approx \frac{1}{N} \sum_i f^*(x_i)$$

- 方差

$$V = \frac{1}{N} \sum_i [f^*(x_i)]^2 - I_0^2 \approx \frac{1}{N}$$

- 根据中心极限定理，方差为0的话算的才是完全准确的。
- 减小方差是核心！

掷点法

- I_0实际上也是在正方形内，点在曲线下的概率。
- 产生两个（0,1）上的均匀随机数ξ_1, ξ_2
- 若$\xi_1 \le f^*(\xi_2)$，则给计数器Nu加1
- 总共产生N对数字，从中获得Nu的值。
- $I_0 \approx \frac{Nu}{N}$

优劣

- 掷点法方差

$$V_2 = p(1-p) = I(1-I)$$

- 对比方差

$$V_2 - V_1 = I(1-I) - \int_0^1 [f(x) - I]^2 dx$$

$$= I - I^2 - \int_0^1 f(x)^2 dx + I^2$$

$$= \int_0^1 f(x)\big(1 - f(x)\big) dx \geq 0$$

平均值法更优

特例

- 蒙卡算法的复杂性可能远超想象，其代码本身的脆弱性也需仔细考量
- 考察下面这个很有代表性的例子：

  定义$I(\gamma)=\int_0^1 dx\, x^\gamma=\frac{1}{\gamma+1}, \gamma>-1$
- 平均值法：产生(0,1)内的均匀分布的随机数x,计算x^γ的均值与方差：

$$\frac{1}{N}\sum_i x_i^\gamma=\langle x^\gamma\rangle$$

$$\frac{1}{N}\sum_i x_i^{2\gamma}=\langle x^{2\gamma}\rangle$$

$$std=\frac{\sqrt{\langle x^{2\gamma}\rangle-\langle x^\gamma\rangle^2}}{\sqrt{N}}$$

代码与结果

intgamma.m

```
function res = intgamma(gam,N)
%INTGAMMA 此处显示有关此函数的摘要
%   此处显示详细说明
temp=rand(1,N);
res(1)=sum(temp.^gam)/N;
temp=sum(temp.^(2*gam))/N;
res(2)=(temp-res(1)^2)^0.5/N^0.5;

end
```

N=10000

γ	$\langle x^{\gamma}\rangle \pm std$	$1/(\gamma+1)$
2	0.3342±0.0030	0.333
1	0.4963±0.0029	0.5
0	1±0	1
-0.2	1.2557±0.0033	1.25
-0.4	1.6769±0.0159	1.667
-0.5	1.9639±0.0239	2
-0.6	2.4890±0.0640	2.5
-0.8	3.9503±0.1296	5
-0.9	5.8357±0.3416	10

跑了三次里的最差结果

由于误差的形式，N提高100倍，小数点更精确一位，故我们确实应该获得精确到小数点后第二位的结果。但是后两个明显有问题：即使跑多次，中心极限定理告诉我们，每次的结果不能偏离正确值太远。问题在哪里？？？

实时平均

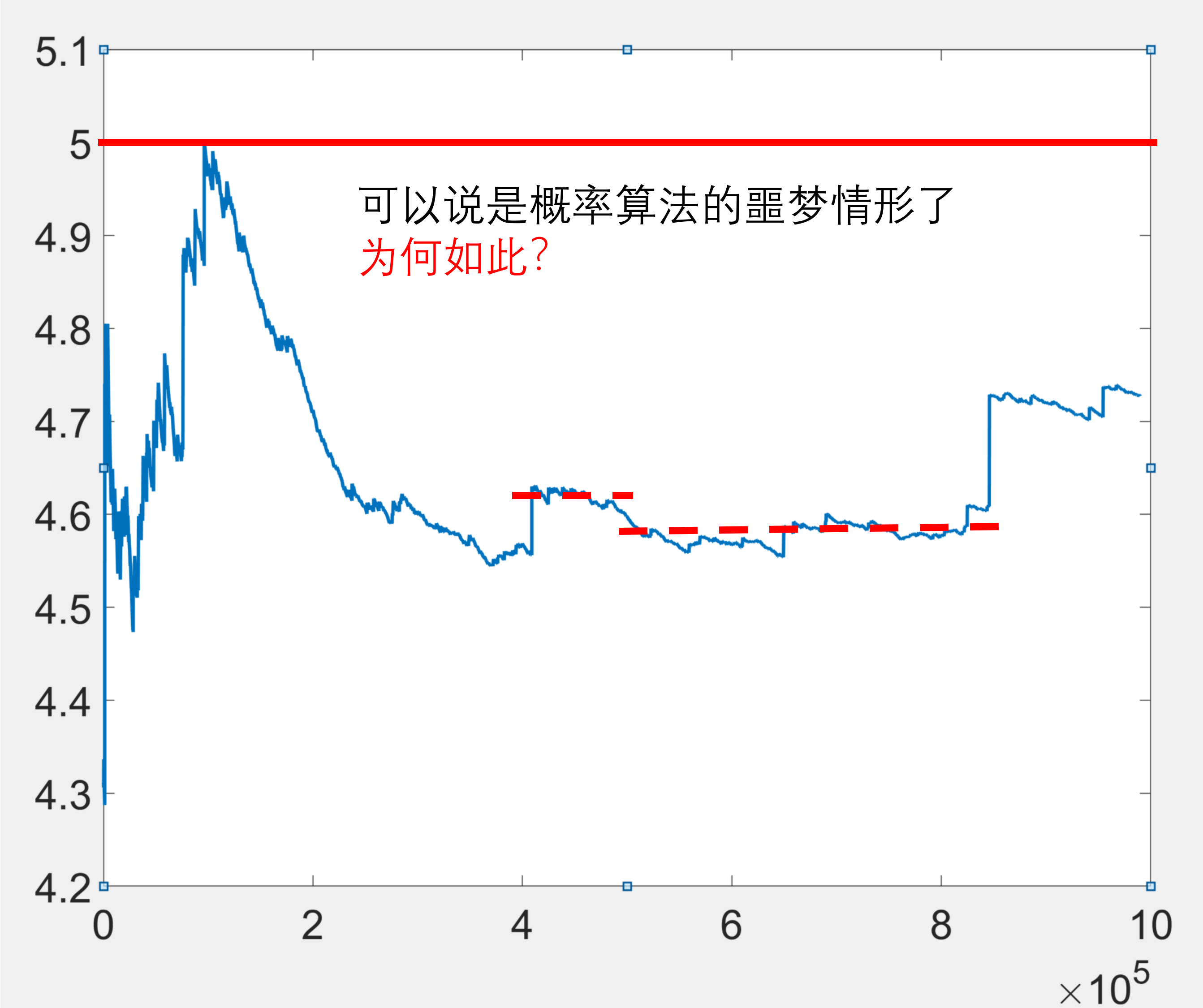
这实际是收敛性检测

intgamma2.m

```matlab
function res = intgamma2(gam,N)
%INTGAMMA 此处显示有关此函数的摘要
%   此处显示详细说明
res=zeros(1,N);

for i=1:N
    temp=rand;
    res(i)=((i-1)*res(i-(i>1.5))+temp^gam)/i;
end

end
```



减小方差1：重要性抽样

- 被积函数在积分域内变化很大，则方差较大，平均值法与掷点法误差都较大。

- 考虑在取值较大的区域多投点，在取值较小的区域少投点。
- 方法：$f(x) = g(x)\frac{f(x)}{g(x)} = g(x)f^*(x)$
- $g(x)$是一个概率密度函数，它满足：（1）好抽样；（2）在$f(x)$大的区域它也大，在$f(x)$小的区域它也小。
- 积分的估计值变为
  $I \approx E\left(f^*(x)\right)$，其中x按照$g(x)$分布。

多维推广

- 对于多维积分，仍然需找到适当的函数$g(\boldsymbol{x})$
- 产生多维的随机向量$\boldsymbol{x} = (x_1, \cdots, x_n)$
- 按照舍选法使得其满足$g(\boldsymbol{x})$的分布
- 有时可能需要做一些变量替换（比如之前讲的多维正态分布）

- 重要性抽样的局限性：

（1）找到满足要求的g(x)可能很难；

（2） 当g(x)在某点很小时，数值计算可能有问题，因为待求期望的是f(x)/g(x)

例子

- 重新考察前面的积分问题

$$I(\gamma) = \int_0^1 dx\, x^\gamma = \frac{1}{\gamma+1}, \gamma > -1$$

- 计算方差

$$\int_0^1 dx\, x^{2\gamma} = \frac{1}{2\gamma+1}, \gamma > -\frac{1}{2}; \infty, \gamma < -\frac{1}{2}$$

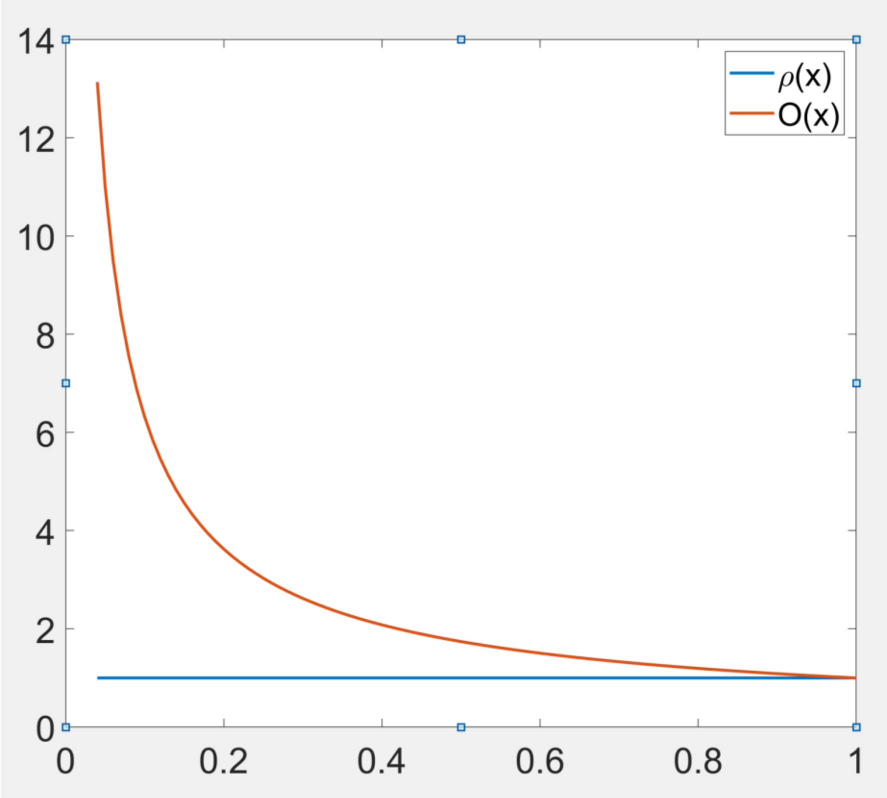
- 用前述的样本的方差无法估计到无穷的真实方差!
- 解决：用重要性抽样方法，在被积函数较大的区域多采样。
- 算法：将被积函数分解$f(x) = O(x)\rho(x)$

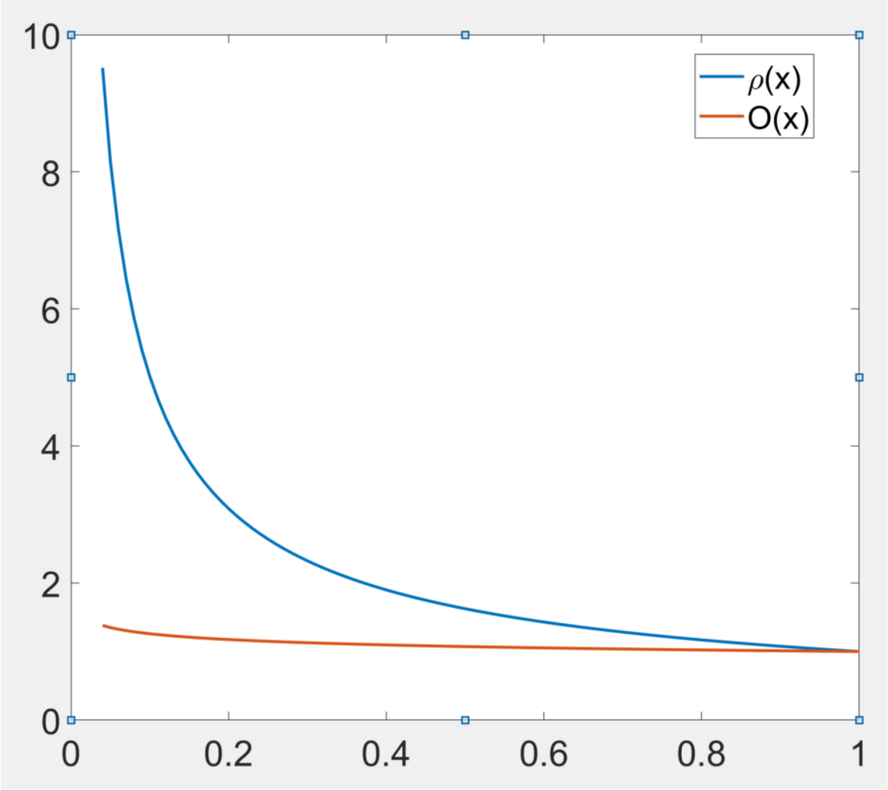
其中$\rho(x) = x^\xi$为概率密度($\gamma < \xi < 0$)），而

$O(x) = x^{\gamma-\xi}$为带求量。

$\langle O(x) \rangle = I(\gamma)$, $\langle O(x)^2 \rangle = \frac{1}{\xi+1} \int_0^1 dx\ x^{2\gamma-\xi}$,只要$2\gamma - \xi > -1$即可

重要性抽样示意





$$\langle O(x)\rangle = \frac{1}{N}\sum_i O_i \approx \frac{\int_0^1 dx\, O(x)\rho(x)}{\int_0^1 dx\, \rho(x)} = \frac{\xi+1}{\gamma+1} = \frac{I(\gamma)}{I(\xi)}$$

$$\langle O(x)^2\rangle = \frac{1}{N}\sum_i O_i^2 \approx \frac{\int_0^1 dx\, O(x)^2\rho(x)}{\int_0^1 dx\, \rho(x)} = (\xi+1)\int_0^1 dx\, x^{2\gamma-\xi},$$

只要$2\gamma-\xi > -1$即可收敛

代码与结果

intgamma3.m

```
function res = intgamma3(gam,xx,N)
%INTGAMMA 此处显示有关此函数的摘要
%   此处显示详细说明
temp=rand(1,N);
t2=temp.^(1/(xx+1));
res(1)=sum(t2.^(gam-xx))/N;
res(2)=(sum(t2.^(2*(gam-xx)))/N-res(1)^2)/N^0.5;
end
```

γ	ξ	$\langle O\rangle \pm std$	$\frac{\xi+1}{\gamma+1}$
-0.4	0.0	1.6785±0.0216	1.66
-0.6	-0.4	1.5058±0.0067	1.5
-0.7	-0.6	1.3237±0.0020	1.33
-0.8	-0.7	1.5135±0.0072	1.5

$$I(-0.8)=\frac{I(-0.8)}{I(-0.7)}\frac{I(-0.7)}{I(-0.6)}\frac{I(-0.6)}{I(-0.4)}\frac{I(-0.4)}{I(0.0)}\;I(0.0)=5.0636$$

根据独立变量乘积的误差传递公式，我们有

$V(O)=\left[\left(\frac{0.0216}{1.6785}\right)^2+\left(\frac{0.0067}{1.5058}\right)^2+\left(\frac{0.0020}{1.3237}\right)^2+\left(\frac{0.0072}{1.5135}\right)^2\right]*5.0636^2=0.0054$， 故$I(-0.8)=5.0636\pm0.0734$ 完全修复了！

质量控制

- 问题引申：我们如何找到一个办法来判断是不是积分会收敛到所需值（也即方差有限）？如何判断积分是否收敛?

- 可用随机游走的原理加以处理

- 为粒子选定一个初始位置，让它每次按一个区间内均匀分布的随机数来行走。
- 用metropolis算法来判定在每步行走中的真实位移，其中用作判断的函数是$f(x)^2$ (在我们的例子中是$x^{2\gamma}$)
- 如果方差不是有限，粒子会被困在某些发散点附近。

原因

- 按照上述方法做随机游走，我们实际上是希望获得按照$f^2(x)$分布的位型x
- 若积分收敛，上述分布做归一化没有问题，我们可以获得此种分布
- 若积分不收敛，归一化系数是1/∞，也即粒子无法处于非发散位置。
- 可将积分区域分段$(0, a_1), (a_1, a_2), (a_2, a_3), (a_3, 1)$。
- 粒子处于每个区间的概率正比于被积函数在相应区间的积分。
- 第一个区间包含发散点，积分为∞，故粒子无法处于其它区间

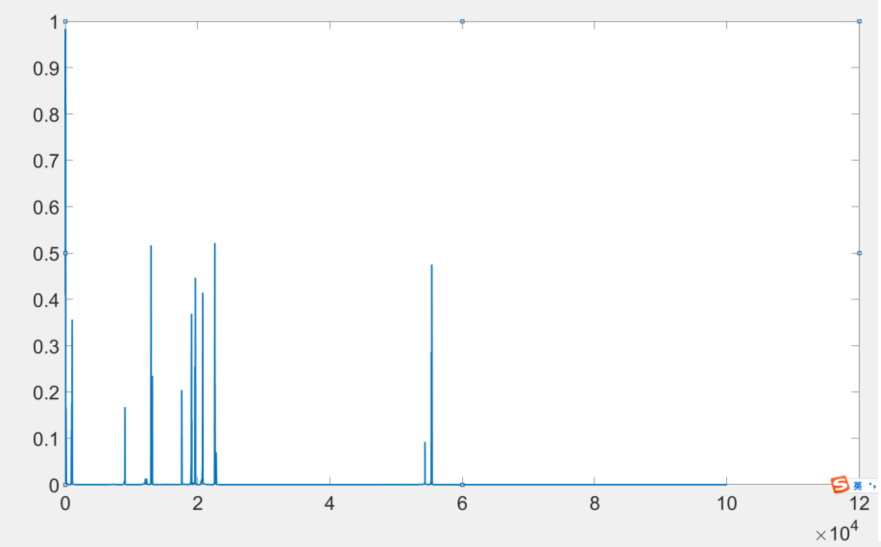
代码与结果

gammawalk.m

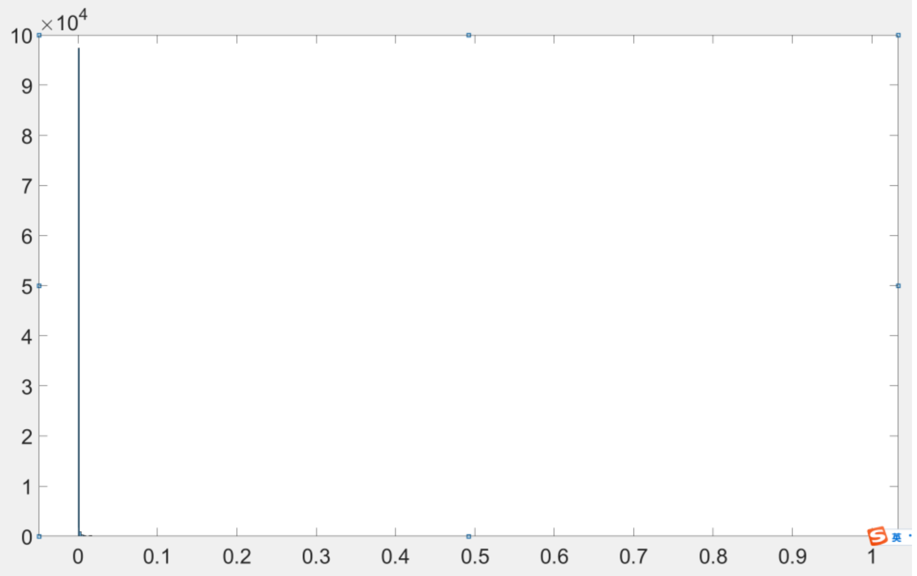
```
function res = gammawalk(x0,delta,N)
%GAMMAWALK 此处显示有关此函数的摘要
%   此处显示详细说明

res=zeros(1,N+1);
res(1)=x0;
gam=-0.8;
for i=2:N+1
    t1=rand;
    t2=(t1-0.5)*delta*2;
    newt=res(i-1)+t2;
    if (newt<0) | (newt>1)
       res(i)=res(i-1);
    else
        lamb=newt^(2*gam)/(res(i-1)^(2*gam));
        t3=rand;
        if t3<lamb
            res(i)=newt;
        else
            res(i)=res(i-1);
        end
    end

end
```



位置变化



分布直方图

减小方差2：分层抽样

- 若投点不均匀，误差会增大；若起伏过大，误差也会增大。
- 伪随机数算法在点数过少时确实均匀性很差（见之前的课件的统计检验结果）
- 适当减小区间可以降低不均匀的问题，而在每个区间，函数起伏一般也会变小，故小区间可能有利降低误差
- 黎曼积分的特性利于区间分划

$$I=\int_0^1 dx f(x)=\int_0^a f(x)dx+\int_a^1 dx\, f(x),\quad 0<a<1$$

减小方差2：做法

- 考察积分（h(x)是某种概率分布）

  $$I=\int_0^1 dx\, f(x)=\int_0^1 dx\, g(x)h(x)$$

- 分层：将区间分成J份（在0与1之间插入J-1个点）

  $$p_j=\int_{x_{j-1}}^{x_j} dx\, h(x)\,,\qquad j=1,2,\cdots J$$

  $h_j(x)=\frac{h(x)}{p_j}$，$x_{j-1}\le x\le x_j$,此为在此区间归一化的概率分布

  $$I_j=\int_{x_{j-1}}^{x_j} dx\, h_j(x)\, g(x)$$

- 合并：则

  $$I=\sum_j p_j I_j$$

- 例子：均匀分布，则$h(x)=1, h_j(x)=\frac{1}{x_j-x_{j-1}}$，$p_j=x_j-x_{j-1}$

减小方差2：如何分层

- 考察积分（h(x)是某种概率分布）

  $I=\int_0^1 dx\, f(x)=\int_0^1 dx\, g(x)h(x)$

- 分层：将区间分成J份（在0与1之间插入J-1个点）

  $p_j=\int_{x_{j-1}}^{x_j} dx\, h(x)\,, \quad j=1,2,\cdots J$

  $h_j(x)=\frac{h(x)}{p_j}$，$x_{j-1}\le x\le x_j$,此为在此区间归一化的概率分布

  $I_j=\int_{x_{j-1}}^{x_j} dx\, h_j(x)\, g(x)$

- 合并：则

$$I=\sum_j p_j I_j$$

- 例子：均匀分布，则$h(x)=1, h_j(x)=\frac{1}{x_j-x_{j-1}}$，$p_j=x_j-x_{j-1}$

减小方差2：执行

- 应在每个区间独立做积分
- 在第j个区间，以$h_j(x)$的概率密度抽样得一系列位型$x_{ji}, i = 1,2,\cdots,n_j$
- 则对第j个积分的估计值为

$$I_j \approx \frac{1}{n_j}\sum_i g(x_{ji})$$

- 随机变量的理论方差为

$$\sigma_j^2 = \int_{x_{j-1}}^{x_j} dx\, h_j(x)\, g(x)^2 - I_j^2$$

- 总随机变量的理论方差为

$$V(I) = V\left(\sum_j p_j I_j\right) = p_j^2 V_j$$

减小方差2：对比

- 对比：若采用均值法按照$h(x)$获得一系列位型，则对积分的估计值为

$$I \approx \frac{1}{N}\sum_i g(x_i)$$

- 此随机变量的理论方差为

$$\sigma_t^2 = \int_0^1 dx\, g(x)^2 f(x) - I^2$$

- 随机变量的理论方差在分层之后的变化

$$\sigma_t^2 = \int_0^1 dx\, [g(x) - I]^2 h(x) = \sum_j \int_{x_{j-1}}^{x_j} dx\, h(x)\, [g(x) - I]^2$$

$$= \sum_j p_j \int_{x_{j-1}}^{x_j} dx\, h_j(x) \left[g(x) - I_j + I_j - I\right]^2$$

$$= \sum_j p_j \int_{x_{j-1}}^{x_j} dx\, h_j(x) \left[\left(g(x) - I_j\right)^2 + \left(I_j - I\right)^2 + 2(g(x) - I_j)(I_j - I)\right]$$

$$= \sum_j p_j \sigma_j^2 + p_j \left(I_j - I\right)^2$$

减小方差2：样本方差

- 样本方差应该采取中心极限定理中的计算方式
- 对于均值法

$V=\frac{\sigma_t^2}{N}$,也即单变量的方差除以总变量数。这里是将积分看成若干个独立同分布的随机变量的和，便于讨论真实值的范围。

- 对于分层法

$$I \approx \sum_j \frac{p_j}{n_j} \sum_i g(x_{ji}),\ V_s = \sum_j \frac{p_j^2}{n_j^2} \sum_i V(x_{ji}) = \sum_j \frac{p_j^2}{n_j} \sigma_j^2$$

- 作差

$$V - V_s = \frac{1}{N} \sum_j \left(p_j \sigma_j^2 + p_j \left(I_j - I\right)^2 \right) - \sum_j \frac{p_j^2}{n_j} \sigma_j^2$$

$$= \sum_j p_j \left(\frac{1}{N} - \frac{p_j}{n_j} \right) \sigma_j^2 + \sum_j \frac{p_j}{N} \left(I_j - I\right)^2$$

- 假设$N = \sum_j n_j$

减小方差2：分层方法

- 在区间分划好之后，每个区间的抽样数应该是多少？
- 可将上式对n_j做变分，也即对下式做变分

$$L = \sum_j p_j \left(\frac{1}{N} - \frac{p_j}{n_j}\right) \sigma_j^2 - \lambda \sum_j n_j$$

- $\frac{\partial L}{\partial n_j} = \frac{p_j^2}{n_j^2}\sigma_j^2 - \lambda = 0 \Rightarrow n_j = \frac{p_j \sigma_j}{\sqrt{\lambda}}$
- 利用归一化$\sum_j n_j = N \Rightarrow \sqrt{\lambda} = \frac{\sum_j p_j \sigma_j}{N}$
- 从而

$$n_j = N * \frac{p_j \sigma_j}{\sum_j p_j \sigma_j}$$

- 此时方差的差值为（这个值越大越好）

$$\sum_j p_j \left(\frac{1}{N} - \frac{p_j}{n_j}\right) \sigma_j^2 = \sum_j \frac{p_j}{N}\left(1 - \frac{\sum_j p_j \sigma_j}{\sigma_j}\right) \sigma_j^2 = \frac{1}{N}\left(\sum_i p_i \sum_j p_j \sigma_j^2 - \left(\sum_i p_i \sigma_j\right)^2\right) \geq 0$$

$$\sum p_i \sigma_i = \sum_i \sqrt{p_i}\ \sqrt{p_i}\sigma_i \leq \sqrt{\left(\sum_i p_i\right)\left(\sum_j p_j \sigma_j^2\right)}$$

减小方差2：特例

- 考察如下情形

$$\frac{p_j}{n_j} = \frac{1}{N}, \ n_j = Np_j$$

　方差差值的第一项为0，由第二项贡献。此时也可减少方差。

- 均匀分布

$$n_j = \frac{N}{J}$$

- 此时也满足前述情况。故均匀分布可使方差变小，但通常不是最优的。

分层抽样：举例

- 考察如下积分

$$I = \int_0^5 dx\, x^3 = \frac{5^4}{4} = 156.25$$

- 被积函数在定义域内变化较大

- 首先用均值法做抽样

$$I = 5 * \int_0^5 dx \ \ x^3 \frac{1}{5}$$

注意：1/5 是均匀分布的概率密度函数

- 然后用分层抽样

$$I = \int_0^1 dx\, x^3 + \int_1^2 dx\, x^3 + \int_2^3 dx\, x^3 + \int_3^4 dx\, x^3 + \int_4^5 dx\, x^3$$

直接抽样部分

directint.m　sint.m　+

```
function res = directint(N)
%DIRECTINT 此处显示有关此函数的
%    此处显示详细说明

x=rand(1,N);
t=x*5;
res=5/N*sum(t.^3);


end
```

N=10000,
参考值：　156.25
运行5次的结果如下

次数	1	2	3	4	5
结果	155.9151	152.6997	156.4348	155.3659	157.5115
偏差	-0.3349	-3.5503	0.1848	-0.8841	1.2615

均匀分层

irectint.m sint.m

```
function res = sint(N)
%SINT 此处显示有关此函数的摘要
%   此处显示详细说明
y=zeros(1,5);
sig=y;
for i=1:5
    x=rand(1,N/5);
    x=x+i-1;
    y(i)=5/N*sum(x.^3);
    sig(i)=(5/N*sum(x.^6)-y(i)^2)^0.5;
end
res(1)=sum(y);
```

N=10000, 每层2000
参考值： 156.25
运行5次的结果如下

次数	1	2	3	4	5
结果	156.7263	156.6907	155.7487	155.9722	156.3783
偏差	0.4763	0.4407	-0.5013	-0.2778	0.1283

实时校正的不均匀分层

irectint.m　sint.m

```matlab
function res = sint(N)
%SINT 此处显示有关此函数的摘要
%   此处显示详细说明
y=zeros(1,5);
sig=y;
for i=1:5
    x=rand(1,N/5);
    x=x+i-1;
    y(i)=5/N*sum(x.^3);
    sig(i)=(5/N*sum(x.^6)-y(i)^2)^0.5;
end
res(1)=sum(y);
```

N=10000, 每层2000
参考值： 156.25
运行5次的结果如下

次数	1	2	3	4	5
均匀情形	156.7263	156.6907	155.7487	155.9722	156.3783
偏差	0.4763	0.4407	-0.5013	-0.2778	0.1283
校正之后	156.6453	155.6427	156.2902	156.3982	156.3497
偏差	0.3953	-0.6073	0.0402	0.1482	0.0997

均值法的结果

次数	1	2	3	4	5
结果	155.9151	152.6997	156.4348	155.3659	157.5115
偏差	-0.3349	-3.5503	0.1848	-0.8841	1.2615

减小方差3：控制变量

- 仍然是找一个与原来函数比较接近的函数g(x)

- 相减而不是相除

$$I = \int_a^b dx\,[f - g] + \int_a^b dx\, g(x)$$

后面的积分需要比较好算

前面的被积函数起伏变小

减小方差4：对偶变量

- 变量之和的方差
  $V(f_1+f_2) = V(f_1) + V(f_2) + 2E((f_1 - \langle f_1 \rangle)(f_2 - \langle f_2 \rangle))$
- 如果两个变量负相关，方差会变小
- 对于单调递增的函数$f(x)$，如果我们需要计算
  $I = \int_0^1 dx\, f(x)$
- 可产生(0,1)上的均匀分布的变量ξ_i，并计算
  $I = \frac{1}{N}\sum_i \frac{[f(x_i)+f(1-x_i)]}{2}$
- f(x)与f(1-x)是明显负相关的。
- 同样的方法也适用于单调递减的函数。

证明

- 均值法的方差为

$$V_1 = \int_0^1 dx\, f(x)^2 - I^2$$

- 对偶法的方差为

$$V_2 = V\left(\frac{f(x)}{2}\right) + V\left(\frac{f(1-x)}{2}\right) + \frac{1}{2}E\big((f(x)-I)(f(1-x)-I)\big)$$

$$= \frac{1}{4}V_1 + \frac{1}{4}V_1 + \frac{1}{2}E\big((f(x)-I)(f(1-x)-I)\big)$$

- 做差 $V_1 - V_2 = \frac{1}{2}\int_0^1 dx\, f(x)^2 - \frac{1}{2}E\big(f(x)f(1-x)\big)$

$$= \frac{1}{2}\sum_i \delta x_i\, f(x_i)^2 - \frac{1}{2}\sum_i f(x_i)\sqrt{\delta x_i}\, f(1-x_i)\sqrt{\delta x_i}$$

$$\geq \frac{1}{2}\sum_i \delta x_i\, f(x_i)^2 - \frac{1}{2}\sqrt{\left(\sum_i f(x_i)^2 \delta x_i\right)\left(\sum_j f\left(1-x_j\right)^2 \delta x_j\right)}$$

$$= \frac{1}{2}\sum_i \delta x_i\, f(x_i)^2 - \frac{1}{2}\sum_i \delta x_i\, f(x_i)^2$$

$$= 0$$

多重积分

- 产生高维随机向量
- 利用分层或者重要性抽样的方法降低方差。
- 与一维积分没有本质区别。

作业

1. 教材第三章第二题中的积分，请用题中所说的重要性抽样法计算此积分（抽样点数为10000）：（1）写出算法过程；（2）写代码进行计算。
2. 仍然是上述积分，限制积分范围为[0,20],请用分层抽样法计算此积分（总抽样点数为10000，任何分层抽样法都可）：（1）写出算法过程；（2）写代码进行计算。