11.1〜11.3 サンプリング法

平成 28 年 8 月 26 日

概 要

PRML の「11.1〜11.3 サンプリング法」についての実装と考察

目 次

1　問題設定

サンプリング法について試す.

2　11.1.1 Box-Muller 法

2.1　アルゴリズム

(0,1) の一様分布から任意のガウス分布に従う乱数を生成するアルゴリズムである.

Box-Muller 法

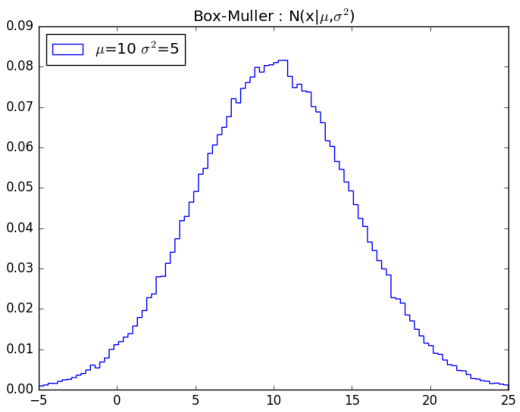
1. $z_1, z_2 \in (0,1)$ の一様分布に対し, $z \to 2z - 1$ とする.
2. $z_1^2 + z_2^2 = r^2 \geq 1$ に対して, $y_i = z_i(-2\ln r^2/r^2)^{1/2}$ とする.

2.2　コード

Box-Muller 法のコード (Box_Muller.py).

```
N=100000
z1=rd.rand(N)
z2=rd.rand(N)
mu,sig=10,5
Y=[]
NUM=0
for n in range(N):
        z1[n]=z1[n]*2-1
        z2[n]=z2[n]*2-1
        r=z1[n]**2+z2[n]**2
        if r<=1:
                y1=z1[n]*sqrt(-2*log(r)/r)
                y2=z2[n]*sqrt(-2*log(r)/r)
                y1=y1*sig+mu
                y2=y2*sig+mu
                Y.append(y1)
                Y.append(y2)
                NUM+=2
```

2.3　結果



3　11.1.2 棄却サンプリング

3.1　アルゴリズム

$p(z) = \frac{1}{Z_p}\tilde{p}(z)$ として, $\tilde{p}(z)$ は容易に求められるが, Z_p はわからないとするときに乱数を生成する方法である. ここで, $kq(z) \geq \tilde{p}(z)$ となるように提案分布 $q(z)$ を導入する.

棄却サンプリング

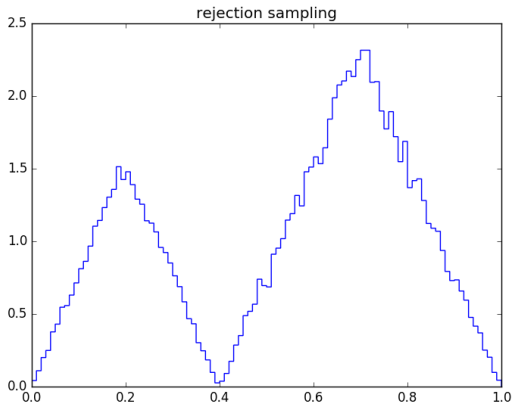
1. 乱数 z_0 を $q(z)$ から生成する.
2. 乱数 y_0 を区間 $[0, kq(z_0)]$ 上の一様分布から生成する.
3. $u_0 \leq \tilde{p}(u_0)$ のとき採用する.

3.2　コード

棄却サンプリングのコード (Rejection_Sampling.py).

```
def q(x):
        if 0<x<=1:
                return 1
def p(x):
        if x<=0.2:
                return x
        elif 0.2<x<=0.4:
                return 0.4-x
        elif 0.4<x<=0.7:
                return -0.4+x
        elif 0.7<x<=1:
                return 1-x
N=100000
z=rd.rand(N)
k=0.3
Y=[]
for n in range(N):
        u=rd.rand()*k*q(z[n])
        if u<p(z[n]):
                Y.append(z[n])
```

3.3　結果



4　11.1.4 重点サンプル

4.1　アルゴリズム

平均を求めるためのアルゴリズムである.

重点サンプリング

1. 棄却サンプリングで採用した乱数 $z^{(l)}$ について, $\tilde{r}_l = \tilde{p}(z^{(l)})/q(z^{(l)})$ を求める.
2. 関数 f の平均は $E[f] \simeq \sum_l w_l f(z^{(l)}),\quad w_l = \tilde{r}_l / \sum_m \tilde{r_m}$

4.2　コード

重点サンプリングのコード (Importance_Sampling.py).

```
def f(x):
        if 0<x<=1:
                return x

N=100000
z=rd.rand(N)
k=0.3
L,r=[],[]
for n in range(N):
        u=rd.rand()*k*q(z[n])
        if u<p(z[n]):
                r.append(p(z[n])/q(z[n]))
                L.append(n)
NUM=len(L)

w=np.zeros(NUM)
sum_r=np.sum(r)
w=r/sum_r

E=0
for l in range(NUM):
        E+=w[l]*f(z[L[l]])

print("E[f]=",E)
```

4.3　結果

$f(z) = z$ とすることで, 分布 $p(z)$ の平均を求めることができ, 3.3 節で用いた分布の平均は, $E[f] = 0.585687285381$ となる.

5　11.2 Metropris アルゴリズム

5.1　アルゴリズム

Metropois アルゴリズム

1. 棄却サンプリングで採用した乱数 $z^{(l)}$ について, $\tilde{r}_l = \tilde{p}(z^{(l)})/q(z^{(l)})$ を求める.
2. 関数 f の平均は $E[f] \simeq \sum_l w_l f(z^{(l)}),\quad w_l = \tilde{r}_l / \sum_m \tilde{r_m}$

5.2 コード

Metroporis アルゴリズムのコード (Metroporis.py).

```
#データ数
NUM=10000
#パラメータ
p=0.5
N=100

def nCr(n,r):
  return factorial(n) / factorial(r) / factorial(n-r)
y=np.zeros(N+1)
y=[nCr(N,k)*(p**k)*((1-p)**(N-k)) for k in range(N+1)]

k=np.zeros(NUM)
k[0]=50
t=range(NUM)

for i in range(NUM-1):
  q1=rd.rand()
  q2=rd.rand()
  a=int((k[i]+1)%(N+1))
  b=int(k[i])
  c=int((k[i]+N)%(N+1))
  if q1<p:
    r=min([1,y[a]/y[b]])
    if q2<r:
      k[i+1]=a
    else:
      k[i+1]=b
  else:
    r=min([1,y[c]/y[b]])
    if q2<r:
      k[i+1]=c
    else:
      k[i+1]=b
```

5.3 結果