课程名称：《风险管理的数学方法》
提交日期：2019年9月24日
姓名：胡庆涛
学号：1901210003

1、债券组合风险刻画

某资产组合包含d只无风险债券，到期日分别为T_i, s时刻的价格分别为$p(s,T_i)$,记到期日为T_i的债券个数为λ_i, $y(s,T)$表示s时刻到期日为T债券的到期收益率有$p(s,T)=e^{-(T-s)y(s,T)}$，用Δ转换时间刻度后，组合在时间t的价值为

$$V_t=\sum_{i=1}^{d}\lambda_i p(t\Delta,T_i)=\sum_{i=1}^{d}\lambda_i exp(-(T_i-t\Delta)y(t\Delta,T_i))$$

记风险因子变化为$X_{t+1,i}=y((t+1)\Delta,T_i)-y(t\Delta,T_i)$故有，

$$\begin{aligned}L_{t+1}&=-(V_t-V_{t+1})\\&=-(\sum_{i=1}^{d}\lambda_i p(t\Delta,T_i)-\sum_{i=1}^{d}\lambda_i p((t+1)\Delta,T_i)\\&=-\sum_{i=1}^{d}\lambda_i p(t\Delta,T_i)()e^{y(t\Delta,T_i)\Delta-(T_i-t\Delta)X_{t+1,i}}-1)\end{aligned}$$

由上式易知，线性损失为：

$$L_{t+1}^{\Delta}=-\sum_{i=1}^{d}\lambda_i p(t\Delta,T_i)(y(t\Delta,T_i)\Delta-(T_i-t\Delta)X_{t+1,i})$$

进一步近似，假定收益率曲线水平，即$y(s+\Delta)=y(s)+\delta$对所有T成立，可得

$$L_{t+1}^{\Delta}=-V_t(y_t\Delta)-\underbrace{\sum_{i=1}^{d}\frac{\lambda_i p(t\Delta,T_i)}{V_t}(T_i-t\Delta)\delta}_{Duration}$$

2、货币远期风险刻画

将远期中的多头理解为持有外币多头头寸

3、风险贷款组合的风险刻画

4、生成t分布随机数---C++

使用Box——Muller算法生成正态分布随机数，即$X=cos(2\pi U_1)\sqrt{-2lnU_2}$,$U_1$、$U_2$是$[0,1]$均匀分布的随机数，则X为标准正态分布的随机数。

证明如下：

X，Y分别服从标准正态分布且相互独立，则联合概率密度为：

$f(x,y)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2+y^2}{2}}$

将其进行极坐标变换，使$X=Rcos(\theta),Y=Rsin(\theta)$
可得R与θ的分布函数分别为：
$F_R（(r)=\int_0^{2\pi}\int_0^r\frac{1}{2\pi}e^{-\frac{R^2}{2}}Rd\theta dR=1-e^{-\frac{r^2}{2}}$
$F_\Theta(\theta)=\int_0^{theta}\int_0^\infty\frac{1}{2\pi}e^{-\frac{R^2}{2}}Rd\theta=\frac{\theta}{2\pi}$
而$F_R(r)$反函数为$R=\sqrt{-2ln(1-Z)}$,且Z服从$[0,1]$均匀分布时，R分布函数为上式
故令$\theta=2\pi U_1, \quad R=\sqrt{-2lnU_2}$代入可得X与Y的表达式.
代码如下：

```
#include <iostream>
#include <random>
#include <cmath>
#include<time.h>
using namespace std;

int main() {
    double pi=3.1415926897932384;
    double u1, u2, x, y, t;

    srand((unsigned)time(NULL));//使用电脑时间作为种子

    //使用Box-Muller算法生成标准正态分布随机数
    u1 = rand() % RAND_MAX / (double)RAND_MAX; //生成（0，1）的随机数
    u2 = rand() % RAND_MAX / (double)RAND_MAX;
    x = sqrt(-2 * log(u1)) * sin(2 * pi * u2); //x为正态分布的随机数
    y = (-2 * log(u1)) * (cos(2 * pi * u2)) * (cos(2 * pi * u2)); //y为卡方分布的
随机数，自由度为1
    //运算得到t（1）分布随机数
    t = x / sqrt(y / 1);
    cout << "t分布的随机数为:" << t << endl;
    return 0;
}
```

5、近似求解Expected Shortfall---python

针对正态分布，使用近似求解，并将n逐渐变大时的结果与精确解比较，验证近似效果。

正态分布精确求解公式为：$ES_\alpha=\mu+\sigma\frac{\phi(\phi(1-\alpha))}{1-\alpha}$

代码如下：

```
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

#定义ES近似计算公式
def func(n):
    l = np.random.normal(mu, sigma, size=n) #生成正态分布随机数
    L = sorted(l, reverse=True) #对随机数进行降序排序
    k = int(n*(1-alpha))
    if k >1: #k大于1时进行求和运算才有意义
        s = np.sum(L[0:k-1])/k
        return s
    return 0

#代入元素值进行计算
```

```python
mu = 0
sigma =2
alpha = 0.99

#列出不同n对应的近似值
es = []
for n in range(1,int(1e5)):
    est = func(n)
    es.append(est)

#精确计算正态分布ES的值
ES = mu + sigma * stats.norm.pdf(stats.norm.ppf(alpha))/(1-alpha)

#作图查看n变大时的近似效果
plt.plot(es)
plt.axhline(y = ES, color='r',linestyle='-')
plt.show()
```

结果生成的图片如下，可见近似效果明显：