9 蒙卡模拟计算美式期权价格 (b)

使用申明*

2021 年 2 月 23 日

目录

1 简介

使用蒙卡模拟计算美式期权价格除了最小二乘法之外，我们也可以使用参数化执行边界的方法，同样由于简单美式看涨期权不会被提前行使，我们这里只考虑美式看跌期权。参数化执行边界是指假设在任一时刻都存在一个执行股价，当股价低于该价格时我们执行期权。

2 参数化执行边界步骤

1. 根据给定参数确定股价离散化变化过程。
$$S(t+\Delta t) = S(t)\, e^{(r-\frac{1}{2}\sigma^2)\Delta t + \sigma\varepsilon\sqrt{\Delta t}}, \quad \Delta t = \frac{T}{M}, \quad \varepsilon \sim N(0,1)\ . \tag{1}$$
2. 按照股价离散化变化过程，抽样出 N 条股价变化路径并保存下来。
3. 确定每条路径末端期权价格。
4. 将每条路径上的期权价格贴现到上一个时间节点，使得此时间节点处每个股价对应一个贴现来的参考期权价格。
5. 将该时刻每条路径上的股价和参考期权价格组成（股价，参考期权价格）对，然后对所有（股价，参考期权价格）对根据其中股价进行由小到大排序。
6. 设变量累计价格 $total = 0$，最大累计价格 $total\_max = 0$，执行边界股价 $cutoff = 0$。由（股价，参考期权价格）对序列末端向前一步步计算，每次将参考期权价格加入 $total$，再减去 $\max(K - 股价, 0)$，然后判断 $total$ 是否大于 $total\_max$，如果大于，则 $total\_max$ 更新为 $total$，$cutoff$ 更新为该对价格中的股价。
7. 遍历完价格对后，所得的 $cutoff$ 即为最优执行边界股价。我们以此更新该时刻每条路径上股价对应参考期权价格。如果股价大于等于 $cutoff$，期权价格保持为参考期权价格，否则期权价格更新为 $\max(K - 股价, 0)$。
8. 重复步骤 4，5，6，7，直到初始时刻。将所得期权价格取平均即为蒙卡模拟计算出的美式看跌期权价格。

*作者不对内容正确性负责。如果您希望使用部分内容作为报告、文章内容等，请您注明内容来源为“金融工程资料小站”网站。

3 步骤 Python 代码实现

```python
import numpy as np

def sample_paths(r, sigma, S_0, T, M, N):
    """ 生成N条长度为M+1的股价随机变化路径。
    """
    paths = list()
    for i in range(N):
        path = [S_0]
        for j in range(M):
            new_price = path[-1]*np.exp((r-0.5*sigma*sigma)*T/M\
                    +sigma*np.sqrt(T/M)*np.random.normal())
            path.append(new_price)
        paths.append(path)

    return paths

def optimal_boundary_eval(S, P, K):
    """ 输入某时刻所有路径上股价S，和对应的参考期权价格P，K为期权执行价格。
        确定美式看跌期权的最佳执行股价为cutoff,
        返回应用此cutoff后的期权价格（和输入股价次序对应），和cutoff。
    """
    length = len(S)
    data_pairs = [[S[i], P[i]] for i in range(length)]
    data_pairs = sorted(data_pairs, key=lambda x: x[0])

    new_P = list()
    total = 0
    total_max = 0
    cutoff = 0
    for i in range(length-1, -1, -1):
        total += data_pairs[i][1]
        total -= max(0, K-data_pairs[i][0])
        if total > total_max:
            total_max = total
            cutoff = data_pairs[i][0]
    for i in range(length):
        if S[i]<cutoff:
            new_P.append(max(K-S[i], 0))
        else:
            new_P.append(P[i])
    return (new_P, cutoff)

def MC_optimal_boundary_Ame_put(r, sigma, S_0, K, T, M, N):
    """ 蒙卡模拟计算美式看跌期权价格主函数。
    """
    put_price = 0
    paths = sample_paths(r, sigma, S_0, T, M, N)
    # 美式看跌期权执行时刻股票价格和期权价格。
    stock_prices = [paths[i][-1] for i in range(N)]
    put_prices = [max(K-stock_prices[i], 0) for i in range(N)]
```

```
    # 初始化最佳执行边界，留作debug用。
    cutoffs = [0]*(M+1)
    cutoffs[-1] = K
    # 由期权期限末端往回递推。
    for i in range(M-1, -1, -1):
        for j in range(N):
            stock_prices[j] = paths[j][i]
            put_prices[j] = put_prices[j]*np.exp(-r*T/M)
        (put_prices, cutoffs[i]) = optimal_boundary_eval(stock_prices, put_prices, K)
    # 对所有路径初始时刻期权价格取平均值。
    for i in range(N):
        put_price += put_prices[i]/N

    return put_price
```

4 计算示例

考虑无风险利率为 0.1，股价波动率为 0.4，股票初始价格为 50，执行时间为 5 个月后，执行价格为 60 的美式看跌期权。我们选择离散化步数 $M = 200$，进行蒙卡模拟抽样 $N = 40000$ 条股价变化路径。计算得该看跌期权价格如下：

```
if __name__ == "__main__":
    r, sigma, S_0, K, T = 0.1, 0.4, 50, 60, 5.0/12
    M, N = 200, 40000
    Ame_put_price = MC_optimal_boundary_Ame_put(r, sigma, S_0, K, T, M, N)
    print("MC American put price: {0:.5f}".format(Ame_put_price))

Output:
    MC American put price: 10.95476
```

5 参考资料

参考文献

[1] 《期权、期货及其他衍生产品》（原书第 9 版）第 27 章，John C. Hull 著，王勇、索吾林译，机械工业出版社，2014.11 。