宋运翔　计算金融　2016141223037

1.采用数学归纳法，假设当$n=k-1$时成立，$V_k(S_k)$只依赖于S_k。之后证明当$n=k$时也成立。

由$V_k=(w_1,\ldots,w_k)=\frac{1}{1+r}(\tilde{p}V_{k+1}(w_1,\ldots,w_{k+1},H)+\tilde{q}V_{k+1}(w_1,\ldots,w_{k+1},T))$

$$\Rightarrow V_k(w_1,\ldots,w_k)\propto\frac{1}{1+r}(\tilde{p}V_{k+1}(S_{k+1})+\tilde{q}V_{k+1}(S_{k+1}))$$

且已知该期权依赖为非路径依赖，因此$V_N(S_N)$参数最高为一次项

$$\Rightarrow V_k(w_1,\ldots,w_n)\propto V_{k+1}(\frac{\tilde{p}+\tilde{q}}{1+r}S_{k+1})$$

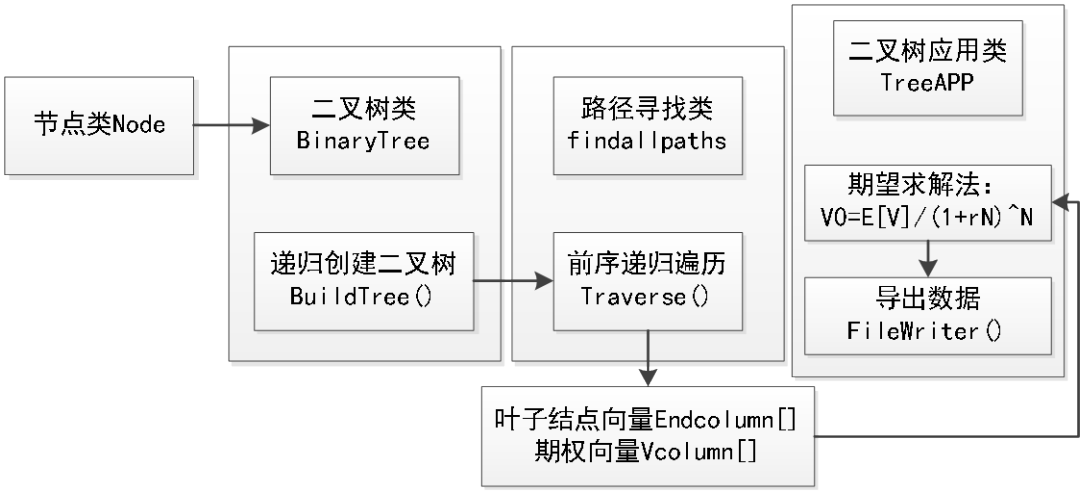
且由于$S_k=\frac{\tilde{p}+\tilde{q}}{1+r}S_{k+1}$

$$\Rightarrow V_k=(w_1,\ldots,w_k)\propto V_{k+1}(S_k)$$

因此可以得到$V_n=(w_1,\ldots,w_n)$只依赖于S_n。

2.编写程序时我们的核心公式为$N\sigma_N^2=\sigma^2t$和$(1+r_N)^N=(1+R)^t$，其中可以发现等式左边为离散型，右边则为连续型，用离散型逼近得到离散型结果，且当N越大，两者的结果更为接近，离散化则是为了方便使用。

3.编写程序时，我采用 JAVA 进行实现，算法流程如下：



其中叶子结点向量即每条路径的最后的股票价格S_n，期权向量指该路径对应的期权价格，通过$V_N=\max_{0\le i\le N}S_i-S_N$计算得出，之后带入期望求解公式得到：

$$V_0=\frac{1}{(1+r)^N}\tilde{E}[V_N]$$

由于 JAVA 绘图并不美观，之后导出数据$N,V_0,Time$，使用 MATLAB 绘制图

像如下：

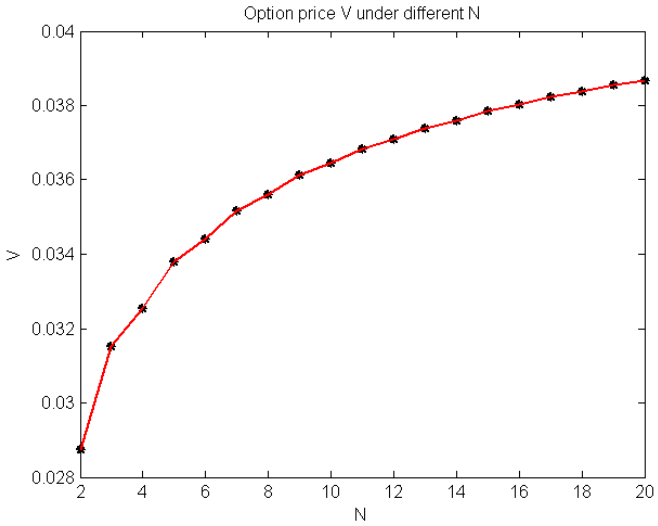


图 3.1 不同 N 时的期权价格 V

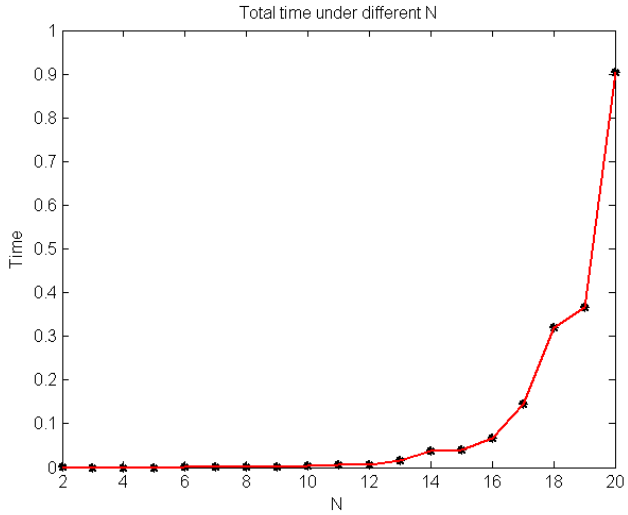


图 3.2 不同 N 时的运算时间 Time

4.假设持有回望期权多头，0 时刻持有资本 X_0，期望在 t 时刻获得收益为：

$$X_0(1+r)^t$$

通过复制期权计算，0 时刻买入股票 Δ_0 对冲空头：

$$\Delta_0 = \frac{V_t(H) - V_t(T)}{S_t(H) - S_t(T)}$$

花费 $\Delta_0 S_0$，剩余 $X_0 - \Delta_0 S_0$ 投资到无风险利率为 r 的货币市场。因此，无论股票上涨还是下跌，在 t 时刻都能获得持有价值为 $X_0(1+r)^t$ 的资产组合。

附录

```
import java.util.*;
import java.lang.*;
import java.io.*;
class Node{
public double data; //节点数据
Node leftChild; //左子节点的引用
Node rightChild; //右子节点的引用
}
class BinaryTree {
Node root;
public BinaryTree() {
this.root = null;
}
public static Node BuildTree(Node node, double S, double u, double d, int N) {
node = new Node();
node.data = S;
//根据步数建立期权二叉树
if (N > 0) {
node.leftChild = BuildTree(node.leftChild, S * u, u, d, N - 1);
node.rightChild = BuildTree(node.rightChild, S * d, u, d, N - 1);
}
return node;
}
}
class findallpaths{
//每条路径叶子结点向量
public ArrayList<Double>Endcolumn= new ArrayList<>();
//每条路径期权向量
public ArrayList<Double>Vcolumn=new ArrayList<>();
public void allpaths(Node root) {
//储存遍历的路径
ArrayList<Double> arrayListSingle = new ArrayList<>();
//先存入根节点
arrayListSingle.add(root.data);
traverse(root,arrayListSingle);
}
//前序递归遍历
public void traverse(Node node,ArrayList<Double>arr){
//遍历左节点
if(node.leftChild != null) {
//在遍历的路径中添加马上开始遍历的左节点,开始遍历左节点
arr.add(node.leftChild.data);
traverse(node.leftChild,arr);
```

```
//遍历完左节点,从遍历的路径中删除左节点
arr.remove(arr.size() - 1);
}
//遍历右节点
if(node.rightChild != null){
//在遍历的路径中添加马上开始遍历的右节点,开始遍历右节点
arr.add(node.rightChild.data);
traverse(node.rightChild,arr);
arr.remove(arr.size() - 1);
}
//当左右节点都空时,说明这个节点是叶子节点
if(node.leftChild == null && node.rightChild == null) {
Endcolumn.add(arr.get(arr.size()-1));
Vcolumn.add(Collections.max(arr)-arr.get(arr.size()-1));
}
}
}
class TreeApp {
public static void main(String[] args) {
//输入参数为现价 S0，连续利率 R，连续标准差 sigma，总步数 N，时间跨度 t
double S0 = 1, R = 0.05, sigma = 0.2, t = 0.0833;
int N;
//定义输出字符串向量
ArrayList<String>plotdata=new ArrayList<>();
for(N=2;N<=20;N++){
//二叉树参数
double TimeA=System.currentTimeMillis();
double sigmaN = Math.sqrt((sigma * sigma * t) / N);
double rN = Math.pow((1 + R), (t / N)) - 1;
double u = Math.exp(sigmaN);
double d = 1 / u;
double p = (1 + rN - d) / (u - d);
//建立期权二叉树
BinaryTree tree = new BinaryTree();
tree.root = BinaryTree.BuildTree(tree.root, S0, u, d, N);
//获得每条路径的期权向量 Vcolumn 和叶子结点向量 Endcolumn
findallpaths find=new findallpaths();
find.allpaths(tree.root);
//计算期权价格
//初始化期权期望价格为 V0
double V=0;
for (int i=0;i<find.Vcolumn.size();i++){
/*M 为下跌的次数，因此 N-M 为上涨的次数
计算方式为 S0*u^(N-M)*d^M=Endcolumn[i]
```

```
两边取对数，化简计算式为(Nlnu-lnEndcolumn[i]+lnS0)/lnu-lnd;
* */
int
M=(int)Math.round((N*Math.log(u)-Math.log(find.Endcolumn.get(i))+Math.log(S0))/(M
ath.log(u)-Math.log(d)));
//每条路径期权为 Vcolumn[I]=p^(N-M)+(1-P)^M
V+=find.Vcolumn.get(i)*Math.pow(p,(N-M))*Math.pow((1-p),M);
}
//期权价格 V0,利用期望求解法，V0=E[V]/(1+rN)^N
double V0=V/Math.pow((1+rN),N);
double TimeB=System.currentTimeMillis();
double Time=(TimeB-TimeA)/1000;
//输出结果
System.out.println("N="+N+"时，期权价格为："+V0+",运算时间为："+Time+"s。");
//将结果添加到输出序列
plotdata.add(N+" "+V0+" "+Time);
}
//考虑到 java 绘图并不美观，将计算数据输出到文本"plotdata.txt"中，使用 matlab 绘图
FileWriter fileWriter = null;
try {
fileWriter = new
FileWriter("D:/Documents/GitHub/First/Financial-stochastic-analysis/plotdata.txt");//创
建文本文件
int i=0;
while(i<19){//循环写入
fileWriter.write(plotdata.get(i)+"\r\n");//写入 \r\n 换行
i++;
}
fileWriter.flush();
fileWriter.close();
} catch (IOException e) {
e.printStackTrace();
}
}
}
```