

4 树形计算可转换债券价格

使用申明*

2021 年 2 月 21 日

目录

1 简介	1
2 计算可转换债券价格步骤	1
3 步骤 Python 代码实现	2
4 计算示例	3
5 参考资料	3

1 简介

可转化债券是一种由公司发行的债券。此种债券的持有人在债券有效期内有权利将债券转换成一定数量该公司的股票，一个单位的债券可转换的股票的数量为转换率。同时这种债券在有效期内一般都是可以被公司以一定价格赎回的，赎回一个单位该债券的价格为赎回价。如果公司将要赎回债券，债券的持有者可以选择将债券转换为股票，即债券被转换的优先级高于债券被赎回。

计算可转换债券的价格时不能忽略公司有可能违约。一种相对简单的计算可转换债券的方式是基于假设公司的股票价格的变化服从几何布朗运动，可以使用一个树形描述股价变化过程，并且在每个 Δt 的时间段内该公司有 $1 - e^{-\lambda \Delta t}$ 的概率会违约。其中 λ 为风险中性违约概率密度。如果公司违约，则在该时刻可转换债券的价格为债券面值乘以回收率。

2 计算可转换债券价格步骤

1. 根据给定参数构建一个公司股价变化的二叉树，其中分叉参数为（由于有概率违约，所以和普通二叉树模型不同）：

$$u = e^{\sqrt{(\sigma^2 - \lambda)\Delta t}}, \quad d = \frac{1}{u}, \quad p_u = \frac{e^{r\Delta t} - de^{-\lambda\Delta t}}{u - d}, \quad p_d = \frac{ue^{-\lambda\Delta t} - e^{r\Delta t}}{u - d}. \quad (1)$$

2. 计算出叶子层每个节点的股价对应的可转换债券的价格 $B_{i,j}$ ，具体为 $B_{i,j} = \max(S_{i,j} \times \text{转换率}, \text{债券面值})$ 。其中 i 为层数， j 为节点位于该层的位置（由低股价到高股价）。
3. 计算前一层每个节点的股价对应的可转换债券的价格。首先计算不考虑债券被转换或赎回的情况，债券价格的期望

$$B''_{i,j} = e^{-r\Delta t} [p_u B_{i+1,j+1} + p_d B_{i+1,j} + p_{\text{违约}} \times \text{面值} \times \text{回收率}], \quad p_{\text{违约}} = 1 - p_u - p_d. \quad (2)$$

如果该价格大于公司可以赎回的价格，那么公司将会赎回，所以

$$B'_{i,j} = \min(B''_{i,j}, \text{赎回价}). \quad (3)$$

*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。

持有者可以选择在公司赎回前转换债券为股票，所以该节点债券最终价格为：

$$B_{i,j} = \max(B'_{i,j}, \text{转换率} \times S_{i,j}) = \max[\min(B''_{i,j}, \text{赎回价}), \text{转换率} \times S_{i,j}] . \quad (4)$$

4. 重复步骤 3，直到计算出根节点处的可转换债券价格。

3 步骤 Python 代码实现

```

import numpy as np 1
E = np.e 2
class Tree_convertible_bond: 3
    def __init__(self, r, sigma, S_0, T, lbd, conversion_ratio, callback_price, par_value, 4
        recycle_ratio, steps): 5
        self.r = r 6
        self.sigma = sigma 7
        self.S_0 = S_0 8
        self.T = T 9
        self.lbd = lbd 10
        11
        self.conversion_ratio = conversion_ratio 12
        self.callback_price = callback_price 13
        self.par_value = par_value 14
        self.recycle_ratio = recycle_ratio 15
        self.steps = steps 16
        17
        self.dt = self.T/self.steps 18
        self.u = E**(((self.sigma*self.sigma-self.lbd)*self.dt)**0.5) 19
        self.d = 1/self.u 20
        21
        self.p_u = (E**(self.r*self.dt)- self.d*E**(-self.lbd*self.dt))/(self.u-self.d) 22
        self.p_d = (self.u*E**(-self.lbd*self.dt)-E**(self.r*self.dt))/(self.u-self.d) 23
        self.p_default = 1-self.p_u-self.p_d 24
        25
        self.bond_price = None 26
        self.tree = None 27
        self.build_tree() 28
        29
    def build_tree(self): 30
        self.tree = list() 31
        for lvl in range(self.steps+1): 32
            row = list() 33
            for j in range(lvl+1): 34
                node = dict() 35
                node["S"] = self.S_0*(self.u**j)*(self.d**(lvl-j)) 36
                node["B"] = None 37
                row.append(node) 38
            self.tree.append(row) 39
        return 40
        41
    def calculate_bond_price(self): 42
        # 简化名称。 43
        tree = self.tree 44

```

r, steps =self.r, self.steps	45
conversion_ratio, callback_price = self.conversion_ratio, self.callback_price	46
recycle_ratio, par_value = self.recycle_ratio, self.par_value	47
dt, u, d = self.dt, self.u, self.d	48
p_u, p_d, p_default = self.p_u, self.p_d, self.p_default	49
	50
# 贴现因子。	51
a = E**(-r*dt)	52
	53
# 边界条件。	54
for node in tree [-1]:	55
node["B"] = max(par_value, node["S"]*conversion_ratio)	56
for lvl in range(steps-1, -1, -1):	57
for j in range(lvl+1):	58
# 由下一层往当前层递推。	59
tree[lvl][j]["B"] = a*p_u*tree[lvl+1][j+1]["B"]+a*p_d*tree[lvl+1][j]["B"]	60
tree[lvl][j]["B"] += a*p_default*par_value*recycle_ratio	61
# 判断是否会被召回。	62
tree[lvl][j]["B"] = min(tree[lvl][j]["B"], callback_price)	63
# 判断是否需要转换。	64
tree[lvl][j]["B"] = max(tree[lvl][j]["B"], conversion_ratio*tree[lvl][j]["S"])	65
	66
self.bond_price = tree[0][0]["B"]	67
return	68

4 计算示例

tree_obj = Tree_convertible_bond(0.05, 0.3, 50, 0.75, 0.01, 2, 113, 100, 0.4, 10)	1
tree_obj.calculate_bond_price()	2
bond_price = tree_obj.bond_price	3
	4
print("可转换债券价格为：{0:.5f}".format(bond_price))	5
	6
Output:	7
可转换债券价格为：106.61156 .	8

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），John C. Hull 著，王勇、索吾林译，机械工业出版社，2014.11 。