Проект по Теории финансов

Оценка американских опционов с использованием биномиальных деревьев ценовой динамики базового актива

17 декабря 2018 г.

Тубденов Виталий, э-301
Московский университет им. М.В.Ломоносова
Экономический факультет

Введение

В проекте на данных компании ПАО "НК "Роснефть"откалибровано дерево, характеризующее динамику цены акции компании, и оценены американские опционы на акцию. Описаны шаги, сделанные для оценки, и код с процедурой калибровки и оценки.

Данные

Для оценки необходимы следующие данные:

1. *Текущая цена акции.* Данные получены на сайте:
   https://www.finam.ru/profile/moex-akcii/rosneft/export/
   Цена закрытия на 04.12.18 оказалась равна 439.00 рублей.

2. *Волатильность цены акции.* Данные о котировках получены на сайте:
   https://www.finam.ru/profile/moex-akcii/rosneft/export/
   Историческая годовая волатильность за 500 дней с 14.12.16 по 04.12.18:

$$\sigma^2_{day} = \sum_{t=1}^{500} \left(ln \frac{P_t}{P_{t-1}} \right)^2$$

$$\sigma = \sqrt{\sigma^2_{day}} * (252^{0.5})$$

   Она была посчитана в Excel *(прилагаю файл)* и оказалась равна 0,23646.

3. *Страйк цена.* Я предполагаю страйк цену равной текущей цене акции.

4. *Длительность опциона.* Я буду оценивать опцион на 90 дней, то есть T=0.247.

5. *Длина шага.* Использую длину шага 1 день, то есть $\Delta t = 0,00274$.

6. *Безрисковая ставка процента.* Данные о безрисковой ставке на 04.12.18 на 3 месяца я нашёл на сайте:
   https://www.moex.com/ru/marketdata/indices/state/g-curve/
   Она оказалась равна 7.48%

Теория

Для применения метода биномиальных деревьев срок действия опциона разбивается на большое количество маленьких временных интервалов длиной Δt. Допустим, что в начале каждого временного интервала цена акции равна S, а в конце - принимает одно из двух возможный значений Su или Sd, где $u > 1$ и $d < 1$. Обозначим вероятность роста цены акции буквой p.

Используется принцип риск-нейтральной оценки, то есть принимаются следующие условия:

1. Ожидаемая доходность всех ценных бумаг, являющихся предметом сделок, зафиксирована на уровне безрисковой процентной ставки.

2. Будущие денежные потоки можно вычислить, применив к их ожидаемым величинам дисконтную ставку, равную безрисковой процентной ставке.

Считая, что эти условия выполняются, построим биномиальное дерево, описывающее изменение цены бездивидендной акции в риск-нейтральных условиях.

Параметры p, u, d должны правильно определять среднее значение и дисперсию изменения цены акции на протяжении временного интервала, длина которого равна Δt. Поскольку мы приняли предположение о риск нейтральном мире, ожидаемая доходность актива равна безрисковой процентной ставке r. Предположим, что доходность актива равна q. В таком случае, ожидаемая капитальная прибыль равна $r - q$. Следовательно, ожидаемое значение цены актива в конце временного интервала протяженностью Δt равно $Se^{(r-q)\Delta t}$. Отсюда следует, что

$$Se^{(r-q)\Delta t} = pSu + (1-p)Sd$$

$$e^{(r-q)\Delta t} = pu + (1-p)d$$

Предположим, что дисперсия относительного изменения цены акции за малый период времени Δt равна $\sigma^2 \Delta t$. В таком случае имеет место равенство:

$$pu^2 + (1-p)d^2 - e^{2(r-q)\Delta t} = \sigma^2 \Delta T$$

В конце концов получаем:

$$e^{(r-q)\Delta t}(u+d) - ud - e^{2(r-q)\Delta t} = \sigma^2 \Delta t$$

Кроме того, в работе *(Cox, Ross, Rubinstein, 1979)* накладывается следующее ограничение:

$$u = d^{-1}$$

Совокупность полученных условий позволяет найти формулы для p, u и d.

$$p = \frac{a - d}{u - d}$$

$$u = e^{\sigma \Delta t}$$

$$d = e^{-\sigma \Delta t}$$

$$a = e^{(r-q)\Delta t}$$

В момент времени $i\Delta t$ цена акции может принимать одно из $i + 1$ значений:

$$S_0 u^j d^{i-j}$$

$$j = 0, 1, ..., i$$

Стоимость опционов вычисляется с помощью обратного обхода дерева, который начинается с одного из узлов дерева, соответствующих моменту T. Стоимость опциона в момент T известна. Например, цена опциона пут в этот момент равна $max(K - S_T, 0)$, а цена опциона колл - $max(S_T - K, 0)$. Поскольку мы приняли предположение о риск-нейтральном мире, стоимость опциона в каждом из узлов дерева, соответствующих моменту $T - \Delta t$ представляет собой стоимость опциона, ожидаемую в момент T, к которой применена ставка дисконта r, соответствующая интервалу продолжительностью Δt. Если опцион является американским, в каждом из узлов необходимо проверить целесообразность дальнейшего владения опционом на протяжении времени Δt. Итак, пройдя все дерево от конца к началу, мы получим стоимость опциона в нулевой момент времени.

Реализация в Python

Метод биномиальных деревьев был реализован мною в Pyhton 3. В приложении приведен код с комментариями. Элементы кода (*stages*):

1. Загрузка пакетов, необходимых для выполнения кода.

2. Определение функции, возвращающей цены опционов, калибровка модели - задание параметров $u, d, p, At = \Delta t$.

3. Построение цен базового актива в узлах биномиального дерева.

4. Нахождение цены опциона на конечных узлах.

5. Процесс нахождение цены опциона на предшествующих узлах - обратная индукция.

6. Входные данные, n - число шагов, S - цена на начало периода, r - безрисковая ставка процента, K - страйк цена, v - волатильность, t - длительность опциона.

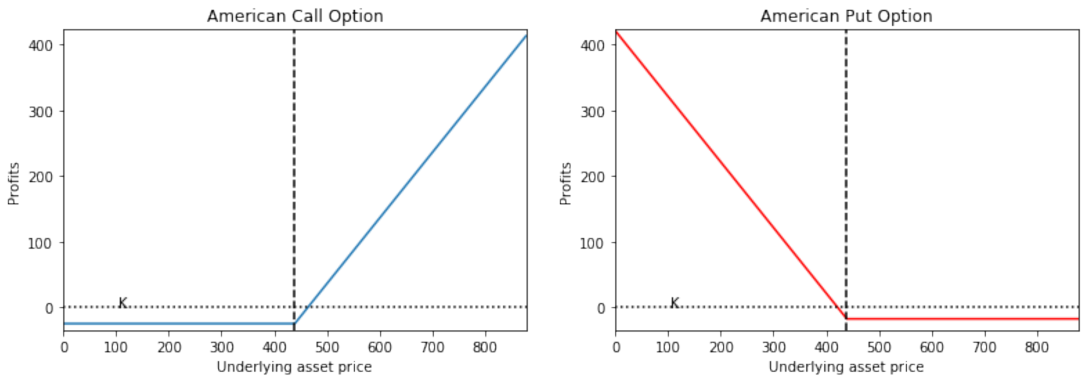
7. Вычисление цен опционов и построение графиков.

Результаты оценки

Модель предсказала цену трехмесячных опционов пут и колл на акции компании ОАО "НК "Роснефть".

Цена опциона колл оказалась равна 24.57 руб. Цена опциона пут оказалась равна 17.31 руб. Графики 1 иллюстрирует результаты оценки модели.

График 1. Результаты модели.



In [ ]:

```
#1 stage
import matplotlib.pyplot as plt
import numpy as np

#2 stage
def Binomial(n, S, K, r, v, t, PutCall):
    At = t/n
    u = np.exp(v*np.sqrt(At))
    d = 1./u
    p = (np.exp(r*At)-d) / (u-d)

#3 stage
    stockvalue = np.zeros((n+1,n+1))
    stockvalue[0,0] = S
    for i in range(1,n+1):
        stockvalue[i,0] = stockvalue[i-1,0]*u
        for j in range(1,i+1):
            stockvalue[i,j] = stockvalue[i-1,j-1]*d

#4 stage
    optionvalue = np.zeros((n+1,n+1))
    for j in range(n+1):
        if PutCall=="C": # Call
            optionvalue[n,j] = max(0, stockvalue[n,j]-K)
        elif PutCall=="P": #Put
            optionvalue[n,j] = max(0, K-stockvalue[n,j])

#5 stage
    for i in range(n-1,-1,-1):
        for j in range(i+1):
                if PutCall=="P":
                    optionvalue[i,j] = max(0,
                                           K-stockvalue[i,j],
                                           np.exp(-r*At)*(p*optionvalue[i+1,j]+
                                                          (1-p)*optionvalue[i+1,j+1]))
                elif PutCall=="C":
                    optionvalue[i,j] = max(0,
                                           stockvalue[i,j]-K,
                                           np.exp(-r*At)*(p*optionvalue[i+1,j]+
                                                          (1-p)*optionvalue[i+1,j+1]))
    return optionvalue[0,0]

#6 stage
n = 90 #
S = 439 #
r = 0.0748 #
K = 439 #
v = 0.236462543 #
t = 90/365 #

#7 stage

y = [-Binomial(n, S, K, r, v, t, "C")] * (K)
y += [x - Binomial(n, S, K, r, v, t, "C") for x in range(K)]
```

```python
plt.plot(range(2*K), y)
plt.axis([0, 2*K, min(y) - 10, max(y) + 10])
plt.xlabel('Underlying asset price')
plt.ylabel('Profits')
plt.axvline(x=K, linestyle='--', color='black')
plt.axhline(y=0, linestyle=':', color='black')
plt.title('American Call Option')
plt.text(105, 0, 'K')
plt.show()

print(Binomial(n, S, K, r, v, t, PutCall="C"))

z = [-x + K - Binomial(n, S, K, r, v, t, "P") for x in range(K)]
z += [-Binomial(n, S, K, r, v, t, "P")] * (K)

plt.plot(range(2*K), z, color='red')
plt.axis([0, 2*K, min(y) - 10, max(y) + 10])
plt.xlabel('Underlying asset price')
plt.ylabel('Profits')
plt.axvline(x=K, linestyle='--', color='black')
plt.axhline(y=0, linestyle=':', color='black')
plt.title('American Put Option')
plt.text(105, 0, 'K')
plt.show()

print(Binomial(n, S, K, r, v, t, PutCall="P"))
```