Обчислення ризику у моделі з активним ринковим часом

Виконала студентка 4-го курсу, спеціальності «Прикладна математика»

Засуха Дар'я Володимирівна

Суть проблеми

Порівняння якості моделей в сенсі обчислення ризику є важливою частиною сьогоднішніх фінансових ринків. Розроблені математичні моделі дозволяють оцінити справедливі ціни, спрогнозувати рух цінних паперів. Для порівняння якостей різних моделей можна застосувати обчислення і порівняння величини ризику.

Модель Блека-Шоулза

- Модель ціни ризикованих активів: $S_t = S_0 e^{\mu t + \sigma W_t}$
- Динаміка ціни: $dS_t = S_t \mu dt + S_t \sigma dW_t$
- Ціна опціону *call* : $C(S,t) = S\Phi(d_1) - Ke^{-rT}\Phi(d_2)$
- Ціна опціону *put* : $P(S,t) = Ke^{-rT}\Phi(-d_2) - S\Phi(-d_1)$

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \; d_2 = d_1 - \sigma\sqrt{T}$$

$\Phi(*)$ - стандартна кумулятивна функція нормального розподілу

Недоліки моделі Блека-Шоулза

1. Розподіл дохідності має важкі хвости
2. Змінна волатильність, кластеризація волатильності
3. Не пристосована для моделювання неліквідності та періодів з нерухомою дохідністю акцій
4. Квадратична дохідність має позитивну кореляцію

Модель з активним ринковим часом

- Модель ціни ризикованих активів: $S_t = S_0 e^{\mu t + \theta T_t + \sigma W_{T_t}}$
- Динаміка ціни: $dS_t = S_t \mu dt + S_t(\theta + \frac{\sigma^2}{2})dT + S_t \sigma dW_{T_t}$
- Ціна опціону *call* : $C(Y, K) = \int_0^\infty \left(P_0 \Phi(d_1) - Ke^{-rY} \Phi(d_2)\right) f_{T_Y}(t) dt$
- Ціна опціону *put : P = call price - stock price + present value of exercise price.*

μ – девіація ($\mu \in R$), σ – дифузія ($\sigma > 0$)

$T_t, t \geq 0$ – випадковий процес, означає «ринковий», «активний» час.

Активний час T_t

Модель з активним ринковим часом

- Додатний, неспадаючий стохастичний процес, з стаціонарними приростами(необов’язково незалежними)
- $T_t = \sum_{i=1}^{[t]} \tau_i + \tau_{[t]+1}(t - [t])$,
- τ_t - це послідовність приростів часу за одиничний проміжок, підкорюється оберненому гамма-розподілу $R\Gamma\left(\frac{\nu}{2}, \frac{\delta^2}{2}\right)$, де $\nu > 2, \delta > 0.$

$f_{R\Gamma}(x, \alpha, \beta) = \frac{\alpha^\beta}{\Gamma(\beta)} x^{-\beta-1} e^{-\frac{\alpha}{x}} I_{x>0}$, де $\alpha = \frac{\delta^2}{2}, \beta = \frac{\upsilon}{2},$ I – індикатор

Option pricing

Апроксимація щільності активного часу

- Ціна опціону *call (справедлива ціна) :*

$$C(Y,K)=\int_0^{\infty}\left(S_0\Phi(d_1)-Ke^{-rY}\Phi(d_2)\right)f_{T_Y}(t)dt$$

де $d_1=\frac{log\frac{S_0}{K}+rY+\frac{1}{2}\sigma^2 t}{\sigma\sqrt{t}}, d_2=\frac{log\frac{S_0}{K}+rY-\frac{1}{2}\sigma^2 t}{\sigma\sqrt{t}}$,

$\Phi(*)$ - стандартна кумулятивна функція нормального розподілу,

$f_{T_Y}(t)$ – апроксимується як щільність випадкової величини:

$$\frac{1}{\sqrt{Y}}f_{R\Gamma}\left(\frac{t-E(\sqrt{Y}-Y)}{\sqrt{Y}},\frac{v}{2},\frac{\delta^2}{2}\right).$$

Розрахунок Option pricing

Початкові задані параметри

- Лог дохідності на основі історичних даних S_t за рік: $X_t = \log(\frac{S_t}{S_{t-1}})$
- ν, μ, δ – параметри розподілу Стьюдента для лог дохідності X_t [1].

```
def calculate_parameters():
    historical_data = pd.read_csv("data_ALL/historical_data.csv")
    log_returns = np.log(historical_data['Last']/historical_data['Last'].shift(1))
    log_returns.dropna(inplace=True)
    nu, mu, delta = t.fit(log_returns)
    return nu, mu, delta
```

- Порахуємо справедливу ціну за нашою моделлю з такими початковими параметрами:

```
S0 = 162.452
K = 160
Y = 5/252
r = 0.05
sigma = 0.89626
```

```
nu = 12.122281271175913
mu = -0.001771446205980524
delta = 3.123328498823122
```

1. Castelli F., Leonenko N. N., Shchestyuk N. Student-like models for risky asset with dependence. *Stochastic Analysis and Applications*. 2017. Vol. 35, no. 3. P. 452–464. URL: https://doi.org/10.1080/07362994.2016.1266945.

Розрахунок Option pricing

Код і отриманий результат

```
def d1_cdf(Y, K, S0, sigma, r, t):
    d1 = (math.log(S0/K) + r*Y + 0.5*sigma*sigma*t)/(sigma*math.sqrt(t))
    return norm.cdf(d1)
```

```
def d2_cdf(Y, K, S0, sigma, r, t):
    d2 = (math.log(S0/K) + r*Y - 0.5*sigma*sigma*t)/(sigma*math.sqrt(t))
    return norm.cdf(d2)
```

```
def fty(Y, nu, delta, t):
    a = (delta*delta)/2
    b = nu/2
    E = (delta*delta)/(nu-2)
    x = (t - E*(math.sqrt(Y) - Y)) / math.sqrt(Y)
    if x <= 0:
        return 0
    else:
        return (math.pow(a,b) / math.gamma(b)) * (x**(-b-1)) * (math.exp(-(a/x)))
```

```
def option_price(Y, K, S0, d1_cdf, d2_cdf, r, nu, delta, sigma):
    def expr(t):
        d1_t = d1_cdf(Y, K, S0, sigma, r, t)
        d2_t = d2_cdf(Y, K, S0, sigma, r, t)
        fty_t = fty(Y, nu, delta, t)
        return ((S0*d1_t - K*math.exp(-r*Y)*d2_t) * fty_t)
    return integrate.quad(expr, 0, np.inf)[0]
```

```
option_pricing = option_price(Y, K, S0, d1_cdf, d2_cdf, r, nu, delta, sigma)
print("Option pricing = " + str(option_pricing))
```

```
Option pricing = 4.042995583891166
```

Обрахована справедлива ціна для call опціону за нашою моделлю = 4.04

Тепер варто порівняти отриманий результат з реальними даними.

Розрахунок Option pricing

Порівняння з реальними даними

- Нижче наведено скріншот з даними для компанії Tesla Inc. з сайту NASDAQ. Бачимо, що ціна call опціону дорівнює 5.20, що відрізняється від раніше обрахованої ціни.
- Отже, користування моделлю несе ризики. Щоб виміряти та оцінити ризики варто використати величину VaR.

TSLA Option Chain

Date	Option	Calls & Puts	Moneyness	Type
May 2023	Composite	Calls & Puts	Near the Money	All (Types)

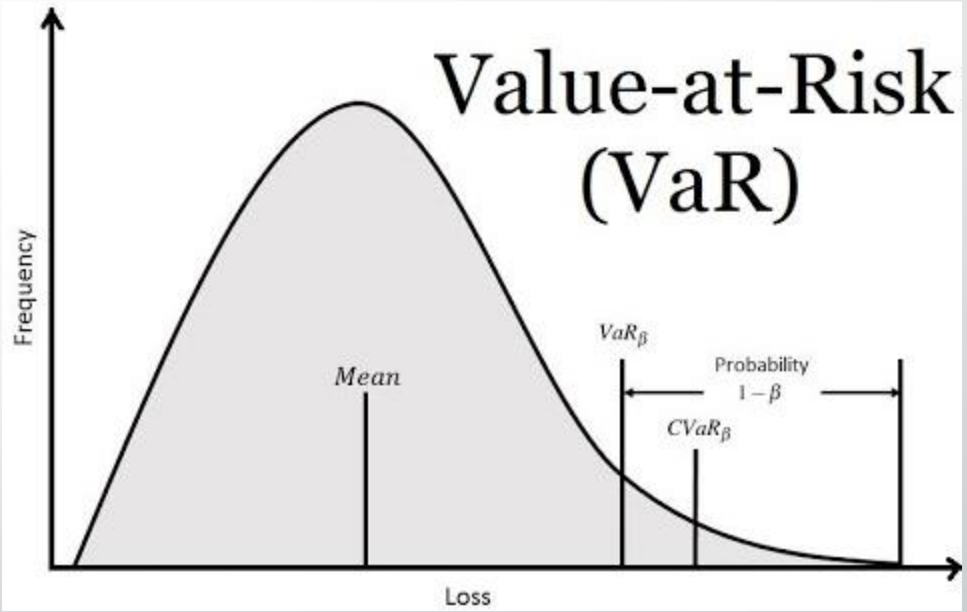
	Calls						
Exp. Date	Last	Change	Bid	Ask	Volume	Open Int.	Strike
May 5, 2023							
May 5	15.90	-1.77 ▼	15.75	16.10	22	111	147.00
May 5	14.81	-1.89 ▼	14.90	15.15	27	260	148.00
May 5	13.75	-2.15 ▼	14.05	14.20	5	496	149.00
May 5	13.39	-1.41 ▼	13.10	13.30	171	2904	150.00
May 5	11.03	-1.47 ▼	10.90	11.05	183	1065	152.50
May 5	8.85	-1.55 ▼	8.80	8.95	984	4783	155.00
May 5	6.98	-1.32 ▼	6.85	7.00	620	4827	157.50
May 5	5.20	-1.40 ▼	5.15	5.30	9315	11576	160.00

Що таке VaR?

- Банки розраховують VaR і оцінюють, чи перевищують вони реалізовані торгові збитки з даною ймовірністю.

- Нехай α буде рівнем довіри, а $\Delta\nu$ буде зміною вартості портфеля до наступного торгового дня. Тоді маємо таке рівняння [2]:

$$P(\Delta\nu \leq -VaR) = \alpha$$



2. Lehar A., Scheicher M., Schittenkopf C. GARCH vs. stochastic volatility: Option pricing and risk management. *Journal of Banking & Finance*. 2002. Vol. 26, no. 2-3. P. 323–345. URL: https://doi.org/10.1016/s0378-4266(01)00225-4.

Метод Монте-Карло для оцінювання VaR

1. Обчислюємо початкову p_t (для сьогоднішнього дня)
2. Симулюємо 1000 S_{t+1} (для завтрашнього дня)
3. Для кожного S_{t+1} з п.2 рахуємо p_{t+1}(для завтрашнього дня)
4. Рахуємо 1000 змін ціни p_{t+1} - p_t, будуємо гістограму (прибутки/втрати)
5. Обчислюємо VaR як 0.95-квантиль розподілу змін ціни з п.4

, де p_t - справедлива ціна call опціону, а S_t - ринкова ціна call опціону

Як знаходимо S_{t+1}?

Метод Монте-Карло

З нашого диференційного рівняння $dS_t = S_t \mu dt + S_t(\theta + \frac{\sigma^2}{2})dT + S_t \sigma dW_{T_t}$ будуємо ітераційну схему:

$$S_{t+1} = S_t + \mu S_t \Delta t + S_t(\theta + \frac{\sigma^2}{2})\tau_t + \sigma\sqrt{\tau_t}\varepsilon_t \quad \text{, де}$$

τ_t - послідовність приростів часу за одиничний проміжок.

ε_t - білий шум із стандартним нормальним розподілом.

Метод Монте-Карло

Код

```
def monte_carlo(n, nu, delta, mu, S0, delta_t, teta, sigma, Y, K, r):
    mu_noise, sigma_noise = 0, 1
    white_noise_S = np.random.normal(mu_noise, sigma_noise, size = n)

    p_t = option_price(Y, K, S0, d1_cdf, d2_cdf, r, nu, delta, sigma)
    print(p_t)

    t_t = invgamma.rvs(scale=((delta*delta)/2), size=n, a=(nu/2))

    S_t_next = np.zeros(len(white_noise_S))
    for i in range(len(white_noise_S)):
        S_t_next[i] = S0 + mu*S0*delta_t + S0*teta*t_t[i] + sigma*math.sqrt(t_t[i])*white_noise_S[i]

    p_t_next = np.zeros(len(white_noise_S))
    for i in range(len(white_noise_S)):
        p_t_next[i] = option_price((Y - delta_t), K, S_t_next[i], d1_cdf, d2_cdf, r, nu, delta, sigma)

    p_t_diff = np.zeros(len(white_noise_S))
    for i in range(len(white_noise_S)):
        p_t_diff[i] = p_t_next[i] - p_t

    mc_table = pd.DataFrame({'S_(t+1)': S_t_next, 'p_(t+1)': p_t_next, 'p_(t+1) - p_t': p_t_diff})

    return mc_table
```

Реалізація обчислення ризику на основі реальних даних

Реальні дані для call опціонів

- У таблиці наведені Strike price, дні та ціни для call опціонів компанії Tesla Inc. Дані взято з сайту NASDAQ.

	Strike	10.02.2023	11.02.2023	12.02.2023	13.02.2023	14.02.2023	15.02.2023	16.02.2023	17.02.2023
0	195.0	16.56	16.51	11.52	8.9	6.55	15.59	19.5	22.1
1	197.5	15.0	14.0	9.23	7.57	5.4	13.54	17.25	19.0
2	200.0	13.45	11.73	8.37	6.52	4.43	11.65	15.1	17.09
3	202.5	11.85	9.98	7.52	5.5	3.6	9.82	12.9	15.49
4	205.0	10.59	8.25	6.5	4.75	2.92	8.25	10.8	13.64
5	207.5	9.35	7.11	5.0	3.98	2.36	6.81	8.95	11.21
6	210.0	8.25	6.21	4.44	3.35	1.92	5.56	7.2	9.99
7	212.5	7.2	5.09	3.75	2.83	1.51	4.5	5.75	8.56

Розрахунок VaR методом Монте Карло

- Нижче наведені значення VaR ($\alpha = 0.95$) для кожної Strike price та кожного дня. Їх було розраховано методом Монте-Карло, а справедливі ціни для кожної Strike price було знайдено за нашою моделлю з активним ринковим часом.

	Strike	10.02.2023	11.02.2023	12.02.2023	13.02.2023	14.02.2023	15.02.2023	16.02.2023	17.02.2023
0	195.0	20.398215	17.439923	16.980837	14.644912	12.582984	9.458606	6.500048	1.738269
1	197.5	19.922578	18.401621	16.14181	14.690004	10.856151	9.873068	5.865228	2.208255
2	200.0	21.11954	17.483676	16.415553	14.256165	11.720501	8.122214	6.023338	2.186244
3	202.5	19.849393	19.385506	16.242464	14.16505	11.763062	9.301298	5.474278	2.636403
4	205.0	19.570994	18.558282	15.347842	13.874757	11.38161	8.272163	6.328363	2.294144
5	207.5	20.460475	17.629859	16.390173	14.665805	11.181655	8.697775	5.26849	1.842774
6	210.0	20.133661	17.222634	15.913366	12.971725	11.349459	8.864418	6.179311	1.66549
7	212.5	20.221994	17.327457	14.998377	13.275748	10.79803	8.193771	6.356446	2.073654

Proportion of failures test

- Тепер необхідно дізнатися чи втрати перевищили знайдене VaR. Для цього знаходимо різниці справедливих цін та цін для call опціонів з реальних даних, а потім порівнюємо з VaR. Отримали бінарну величину, де 0 позначає випадки, коли втрати перевищили VaR.

```
for j in range(strike_count):
    p0 = option_price(Y, var_table['Strike'][j], S0, d1_cdf, d2_cdf, r, nu, delta, sigma)
    for i in range(days_count):
        VaR = var_table.at[j, var_table.columns[i + 1]]
        p_diff = option_table.at[j, option_table.columns[i + 1]] - p0
        if (abs(p_diff) > abs(VaR)):
            failures_table.at[j, failures_table.columns[i + 1]] = 0
        else:
            failures_table.at[j, failures_table.columns[i + 1]] = 1

failures_table
```

	Strike	10.02.2023	11.02.2023	12.02.2023	13.02.2023	14.02.2023	15.02.2023	16.02.2023	17.02.2023
0	195.0	1	1	1	1	1	1	0	0
1	197.5	1	1	1	1	1	1	0	0
2	200.0	1	1	1	1	1	1	0	0
3	202.5	1	1	1	1	1	1	0	0
4	205.0	1	1	1	1	1	1	1	0
5	207.5	1	1	1	1	1	1	1	0
6	210.0	1	1	1	1	1	1	1	0
7	212.5	1	1	1	1	1	1	1	0

Proportion of failures test
Тест-статистика

- Тестову статистику для отриманої бінарної величини обчислимо за формулою [2]:

$$LR = -2\log\left((1-p^*)^{(n-x)}p^{*\,x}\right) + 2\log\left(\left(1-\frac{x}{n}\right)^{(n-x)}\left(\frac{x}{n}\right)^{x}\right)$$

де p^* – порогова ймовірність за нульовою гіпотезою, n – розмір вибірки, x – кількість випадків, коли втрати перевищують VaR (к-ть 0).

```
prob = 0.05
n = days_count*strike_count
x = 0
for j in range(strike_count):
    for i in range(days_count):
        if failures_table.at[j, failures_table.columns[i + 1]] == 0:
            x += 1
LR = -2*math.log(((1 - prob)**(n - x))*(prob**x)) + 2*math.log(((1 - (x/n))**(n - x))*((x/n)**x))
print("Test statistics = " + str(LR))

p_value = 1 - chi2.cdf(LR, 1)
print("p-value = " + str(p_value))
```

```
Test statistics = 15.462148838943499
p-value = 8.417412780192812e-05
```

2. Lehar A., Scheicher M., Schittenkopf C. GARCH vs. stochastic volatility: Option pricing and risk management. *Journal of Banking & Finance*. 2002. Vol. 26, no. 2-3. P. 323–345. URL: https://doi.org/10.1016/s0378-4266(01)00225-4.

Proportion of failures test

Гіпотези та результат

H_0 - частка невдач (випадків, коли втрати перевищили VaR) $\leq \ p^*$

H_1 - частка невдач (випадків, коли втрати перевищили VaR) $> p^*$

, де p^* – порогова ймовірність

Для порогової ймовірності 0.05 отримали p-value = 0.000084, що менше за рівень значущості $\alpha = 0.05$. Отже, маємо підстави для відкидання нульової гіпотези. Це означає, що за даною моделлю з активним ринковим часом частка випадків, коли реалізовані втрати перевищують значення VaR, скоріш за все буде перевищувати порогову ймовірність p^*(тобто 5%).

Proportion of failures test

Додаткові тести

Варто провести додаткові тести з різними значеннями порогової ймовірності. Нижче наведено таблицю значень тест-статистики та p-value для різних значень порогової ймовірності.

Null hypothesis probability	LR	p-value
0.05	15.462149	0.000084
0.10	4.449608	0.034909
0.11	3.324186	0.068268
0.12	2.411067	0.120480
0.15	0.662920	0.415531
0.20	0.063511	0.801030

Отже, ми не маємо підстав для відкидання нульової гіпотези, де p^* дорівнює 11%.

Можемо сказати, що частка випадків, коли реалізовані втрати перевищують значення VaR, за даною моделлю з активним ринковим часом, скоріш за все не буде перевищувати порогову ймовірність p^*, тобто 11%.

Висновки

Для реальних даних call опціонів компанії Tesla Inc. було:

- ✓ знайдено параметри розподілу Стьюдента для лог дохідності X_t
- ✓ апроксимовано щільність активного ринкового часу T_t
- ✓ пораховано справедливі ціни call опціонів для різних Strike цін
- ✓ методом Монте-Карло знайдено значення VaR для різних днів та Strike цін
- ✓ введено та обчислено тест-статистику для бінарної величини

Дякую за увагу!