Opcje azjatyckie warunkowe - metoda Monte Carlo

2022-12-15

Wstęp

Projekt jest ćwiczeniem z wyceny opcji metodą Monte Carlo, na przykładzie opcji azjatyckich typu warunkowego. Opis tych opcji podajemy za [2].

...

```
# Parametry wspólne:
r = 0.05; S_0 = 2; K = 2;
b = 1 # dla opcji warunkowych
# Parametry zmienne:
n = 1000 # długość trajektorii
m = 1000 # ilość trajektorii (prób)
```

Wzór analityczny dla opcji azjatyckiej ze średnią geometryczną

Zakładając, że zmienna losowa X ma rozkład logarytmiczno-normalny, tj. $\log(X)$ ma rozkład normalny $N(m, s)$, wyliczymy $E(X - K)_+$ oraz $E(K - X)_+$ (p. [3]).

...

```
expected_LogNormal = function(m,s) { # EX

}

expected_LogNormal_greater = function(m,s,k) { # E(X; X>k) = E( X 1_{X>k} )

}

p_LogNormal = function(x,m,s) { # F_X(x) = P(X <= x)

}

call_option_on_LogNormal = function(m,s,k) { # E(X-K)_+ = ...

}

put_option_on_LogNormal = function(m,s,k) { # E(K-X)_+  (używamy 'put-call parity')

}
```

Parametry m, s rozkładu zmiennej X wyznaczamy następnie, jak w [1], ss.99-100 ("Geometric average options").

```
expected_payoff_AsianGeomMean_call = function() { # wg [1], 3.2.2
  m = log(S_0) + (r - sigma^2/2) * T * (n+1) / (2*n)
  s = sigma / n * sqrt(T*(n+1)*(2*n+1)/6)
```

```
  return ( call_option_on_LogNormal(m,s,K) )
}

expected_payoff_AsianGeomMean_put = function() { # wg [1], 3.2.2
  m = log(S_0) + (r - sigma^2/2) * T * (n+1) / (2*n)
  s = sigma / n * sqrt(T*(n+1)*(2*n+1)/6)
  return ( put_option_on_LogNormal(m,s,K) )
}
```

Wypłata z opcji azjatyckiej ze średnią geometryczną - Monte Carlo

```
geom_mean_but_first = function(y) {
  return ( exp(mean(log(y[-1]))) ) # śr. geom., bez 1-go elementu (S_0)
}

payoff_AsianGeomMean_call = function(trajectory) {
  return ( max(0, geom_mean_but_first(trajectory) - K) ) # opcja kupna
}

payoff_AsianGeomMean_put = function(trajectory) {
  G_T = exp(mean(log(trajectory[-1])))
  return ( max(0, K - geom_mean_but_first(trajectory)) ) # opcja sprzedaży
}
```

Wycena opcji azjatyckiej ze średnią arytmetyczną, z redukcją wariancji

Korzystamy z metody redukcji wariancji przy pomocy tzw. *control variate*, podanej w [1], ss. 185-187.

```
sample_trajectory_of_S = function() {
  alpha = (r - sigma^2/2)*h
  beta = sigma * sqrt(h)
  S = c(S_0)
  for (i in 1:n) {
    Z = rnorm(1)
    S[i+1] = S[i] * exp(alpha + beta * Z)
  }
  return ( S ) # wektor o długości n+1
}

trapezoidal = function(y) { # całka metodą trapezów na węzłach równoodległych
  s = sum(y) - 0.5 * y[1] - 0.5 * y[n+1]
  return (h * s)
}

arith_mean = function(y) {
  return ( 1/T * trapezoidal(y) )
}

payoff_Asian_call = function(trajectory) {
  return ( max(0, arith_mean(trajectory) - K) ) # opcja kupna
```

```
}

payoff_Asian_put = function(trajectory) {
  return ( max(0, K - arith_mean(trajectory)) ) # opcja sprzedaży
}

estimator_with_control_variate = function(Y_function, X_function, EX) {
  x = c()
  y = c()
  for (i in 1:m) {
    S = sample_trajectory_of_S()
    x[i] = X_function(S)
    y[i] = Y_function(S)
  }
  b = cov(x,y)/var(x)
  yb = y - b * (x - EX)
  return ( mean(yb) )
}
# aktualne parametry:
sigma = .5; T = 1; h = T / n
# aktualne funkcje i wartość oczekiwana:
Y = payoff_Asian_call
X = payoff_AsianGeomMean_call
EX = expected_payoff_AsianGeomMean_call()
# wycena:
price = exp(-r*T) * estimator_with_control_variate(Y, X, EX)
print(price)
```

```
## [1] NaN
```

Wycena azjatyckiej opcji 'put' ze średnią arytmetyczną, z redukcją wariancji

```
# aktualne parametry:
sigma = .5; T = 5; h = T / n
# aktualne funkcje i wartość oczekiwana:
Y = payoff_Asian_put
X = payoff_AsianGeomMean_put
EX = expected_payoff_AsianGeomMean_put()
# wycena:
price = exp(-r*T) * estimator_with_control_variate(Y, X, EX)
print(price)
```

```
## [1] NaN
```

Wycena azjatyckiej opcji warunkowej 'put', z redukcją wariancji

Dla danej trajektorii S, odpowiadająca jej wypłata warunkowej opcji azjatyckiej wygląda tak:

```
payoff_Asian_conditional_call = function(trajectory) {

}
```

```r
payoff_Asian_conditional_put = function(trajectory) {

}
# aktualne parametry:
sigma = .5; T = 5; h = T / n
# aktualne funkcje i wartość oczekiwana:
Y = payoff_Asian_conditional_put
X = payoff_AsianGeomMean_put
EX = expected_payoff_AsianGeomMean_put()
# wycena:
# price = exp(-r*T) * estimator_with_control_variate(Y, X, EX)
# print(price)
```

Konstrukcja azjatyckiej opcji warunkowej - wykres cen i średniej warunkowej

(p. [2], Figure 1)

```r
#plot(t,S,type="l")
#lines(t,S_running_avg)
#lines(t,b_const)
```

Porównanie wyników z literaturą

(p. [2], str. 19, Table 2; kolumny: sigma, AP_b, AP_0)

Otrzymane wyniki, dla różnych wartości sigma oraz m,n, podajemy poniżej, wraz z czasem obliczeń. Obliczenia dokonona zostały na komputerze … Załączamy też, dla porównania, wyniki dostępne w literaturze [2] (Table 2).

```r
sigma_values = .1 * c(6,5,4,3,2)
for (sigma in sigma_values) {

}
```

Literatura:

[1] Paul Glasserman, Monte Carlo Methods in Financial Engineering, Springer 2003

[2] Runhuan Feng, Hans W. Volkmer, Conditional Asian Options, https://arxiv.org/abs/1505.06946v1

[3] https://en.wikipedia.org/wiki/Log-normal_distribution