Dokumentacja

Kalibracja parametrów modelu Hestona za pomocą rozszerzonego filtra Kalmana

Mikołaj Bińkowski
Wiktor Gromniak

Spis treści

1 Wstęp

Celem tego projektu było napisanie szeregu funkcji dla oprogramowania Octave mających za zadanie kalibrację parametrów modelu Hestona do danych cen waniliowych opcji europejskich. Funkcjonalność uzyskano wykorzystując rozszerzony filtr Kalmana. Niniejszy dokument stanowi dokumentację dla kodu Octave.

Postawy teoretyczne oraz motywacja dla zastosowanych rozwiązań zostały zaczerpnięte z prac Lindströma et al. [1] oraz Moodley'a [2].

2 Struktura katalogów

Projekt posiada następującą strukturę katalogów:

```
| calibration
|- data
|- doc
|- filter
|- pricing
```

Poszczególne katalogi zawierają:

- `calibration` - główny katalog projektu; zawiera skrypt z zależnościami `requirements.m` oraz skrypt testujący całą funkcjonalność programu `test.m`
- `data` - zawiera pomocniczą funkcję `data_generator.m` służącą do generowania sztucznych danych testowych oraz same dane w pliku `artificial_data.txt`
- `doc` - zawiera ten dokument, jego źródła oraz literaturę zamieszczoną w bibliografii
- `filter` - zawiera funkcję `extended_kalman_filter.m` realizującą funkcjonalność rozszerzonego filtra Kalmana oraz jej dekorator - funkcję `heston_calibr_kalman.m` dla potrzeb kalibracji modelu Hestona
- `pricing` - zawiera funkcję `heston_call_fft.m` dokonującą wyceny opcji za pomocą modelu Hestona oraz skrypt ją testujący `heston_pricing_test.m`

3 Zależności

Projekt korzysta wyłącznie ze standardowych funkcji pakietu Octave. Nie testowano na Matlabie.

4 Funkcje

Projekt zawiera trzy funkcje realizujące właściwą logikę programu:

```
heston_calibr_kalman
extended_kalman_filter
heston_call_fft
```

Poniżej znajduje się opis poszczególnych funkcji. Wewnętrzna logika każdej z nich opisana jest szczegółowo w kodzie programu wraz z odniesieniami do literatury.

4.1 heston_calibr_kalman

Jest to jedyna funkcja, którą wywołuje użytkownik końcowy programu. Ma ona za zadanie optymalizację parametrów modelu Hestona do danych cen europejskich opcji kupna.

Do optymalizacji wykorzystywany jest rozszerzony filtr Kalmana realizowany przez funkcję `extened_kalman_filter`. Jest ona wywoływana wewnątrz funkcji `heston_calibr_kalman` z odpowiednimi parametrami.

Podstawy teoretyczne stanowi praca Lindströma [1].

4.1.1 Parametry funkcji

- `obs` - macierz obserwacji danych cen opcji o wymiarach $n \times m$, gdzie n jest liczbą obserwacji w czasie, a m jest liczbą opcji w każdym momencie czasu; cena opcji w miejscu (i, j) odpowiada i-tej obserwacji i j-tej wartości `Ts` i `strikes` (patrz niżej)
- `Ts` - wektor czasów do wygaśnięcia długości m; czas do wygaśnięcia liczony jest od pierwszej obserwacji (cf. `ts` poniżej)
- `strikes` - wektor strike'ów o długości m
- `ts` - wektor kolejnych punktów w czasie, w których dokonane były obserwacje, długości n; pierwszy element jest równy 0
- `m_init` - wektor długości 5 zawierający startowe wartości parametrów (te od których zaczną się poszukiwania); kolejne współrzędne odpowiadają: `m(1)~kappa, m(2)~theta, m(3)~s` (opis poszczególnych parametrów - patrz dokumentacja funkcji `heston_call_fft`)
- `r` - stopa procentowa
- `S0` - wektor cen początkowych aktywa bazowego w poszczególnych momentach czasu długości n
- `jac_rec` - parametr opcjonalny; ustala co jaką liczbę iteracji w filtrze Kalmana dokonuje się ponownego obliczenia jakobianu funkcji zadającej proces dynamiczny; pozwala na znaczne skrócenie obliczeń bez istotnej utraty dokładności otrzymanego wyniku; wartość domyślna ustala liczbę przeliczeń na 10 (patrz opis funkcji `extended_kalman_filter` oraz jej kod)

4.1.2 Zwracany wynik funkcji

Funkcja zwraca wektor zoptymalizowanych parametrów. Odpowiedniość pomiędzy współrzędnymi wektora, a parametrami jest taka, jak dla wektora wartości początkowych `m_init` (patrz Parametry funkcji powyżej).

4.2 extended_kalman_filter

Funkcja realizująca rozszerzony filtr Kalmana. Jest zaimplementowana w pełnej ogólności (nie ogranicza się do rozważanego problemu), więc może być wykorzystywana przy innych zagadnieniach.

4.2.1 Podstawy teoretyczne

Podstawy teoretyczne dla implementacji podajemy za Lindströmem [1]. Załóżmy, że interesujący nas układ dynamiczny można zapisać jako

$$y_t = h(x_t) + \varepsilon_t \tag{1}$$
$$x_t = f(x_{t-1}, \eta_t). \tag{2}$$

i załóżmy, że rozkład początkowy x_0 jest Gaussowski ze średnią m_0 i macierzą kowariancji P_0 oraz, że $\varepsilon \sim N(0, R)$ i $\eta \sim N(0, Q)$. Wówczas rozszerzony filtr Kalmana jest zadany jako

1. **Losowanie** x_0 zgodnie z jego rozkładem.
2. **Propagacja** średniej i macierzy kowariancji zadana przez

$$m_{t+1|t} = f(m_t, 0) \tag{3}$$
$$P_{t+1|t} = F(m_t, 0)P_{t|t}F^T(m_t, 0) + G(m_t, 0)QG^T(m_t, 0) \tag{4}$$

3. **Aktualizacja** średniej i macierzy kowariancji zadana przez

$$K = P_{t+1|t}H^T(m_{t+1|t})(H(m_{t+1|t})P_{t+1|t}H^T(m_{t+1|t}) + R)^{-1} \tag{5}$$
$$m_{t+1|t+1} = m_{t+1|t} + K(y_{t+1} - h(m_{t+1|t})) \tag{6}$$
$$P_{t+1|t+1} = P_{t+1|t} - KH(m_{t+1|t})P_{t+1|t} \tag{7}$$

i powtarzaj propagację i aktualizację do momentu gdy $t = T$. [1]

[1] W równaniu (7) $H(m_{t+1|t})$ nie powinno być transponowane (tak jest u Lindströma)!

gdzie

$$F(m,\eta) = \frac{\partial f}{\partial x}(m,\eta) \tag{8}$$

$$G(m,\eta) = \frac{\partial f}{\partial \eta}(m,\eta) \tag{9}$$

$$H(m) = \frac{\partial h}{\partial x}(m) \tag{10}$$

4.2.2 Parametry funkcji

- `y` - macierz obserwacji obserwowalnego procesu dynamicznego y o wymiarach $n \times m$, gdzie n jest liczbą obserwacji w czasie, a m jest liczbą współrzędnych procesu (cf. równ. (1))
- `m_init` - wektor wartości początkowych poszukiwanych parametrów długości k
- `h` - funkcja zadająca wartość procesu y dla zadanych wartości parametrów (cf. równ. (1))
- `H` - pochodna `h` po parametrach (jakobian) (cf. równ. (10))
- `f` - funkcja zadająca relację przejścia dla parametrów (cf. równ. (2))
- `F` - pochodna `f` po parametrach (jakobian) (cf. równ. (8))
- `G` - pochodna `f` po procesie szumu (jakobian) (cf. równ (9))
- `R` - macierz kowariancji szumu dla procesu obserwowalnego o wymianach $m \times m$ (cf. równ. (1))
- `Q` - macierz kowariancji szumu dla procesu parametrów o wymianach $k \times k$ (cf. równ. (2))
- `P0` - wartość początkowa macierzy kowariancji dla procesu parametrów o wymiarach $k \times k$
- `jac_rec` - parametr, który ustala co jaką liczbę iteracji w filtrze Kalmana dokonuje się ponownego obliczenia funkcji H; pozwala na znaczne skrócenie obliczeń bez istotnej utraty dokładności otrzymanego wyniku (patrz kod funkcji)

4.2.3 Zwracany wynik funkcji

Funkcja zwraca wektor zoptymalizowanych parametrów.

4.3 heston_call_fft

Funkcja obliczająca wartość europejskiej opcji kupna w modelu Hestona za pomocą szybkiej transformaty Fouriera (FFT). Dzięki wykorzystaniu FFT jest w stanie robić to bardzo szybko, czego potrzebujemy, ponieważ w zadaniu kalibracji wywołujemy ją wielokrotnie.

Dla ustalenia oznaczeń, rozważamy model Hestona postaci

$$
\begin{aligned}
dS_t &= rS_t dt + \sqrt{V_t} S_t dW_t^1 \\
dV_t &= \kappa(\theta - V_t)dt + \sigma\sqrt{V_t} dW_t^2 \\
d <W_t^1, W_t^2 >&= \rho dt
\end{aligned}
$$

Podstawy teoretyczne dla implementacji stanowi praca Moodley'a [2]. Powyższa postać rozważanego modelu jest zgodna z tą pracą (cf. równ. 1.1, 1.2, 1.3 [2]).

4.3.1 Parametry funkcji

- `kappa` - współczynnik powrotu do średniej, odp. κ
- `theta` - długoterminowa wariancja, odp. θ
- `sigma` - zmienność zmienności, odp. σ
- `rho` - korelacja pomiędzy procesami W_t^1, W_t^2, odp. ρ
- `T` - czas do wygaśnięcia (przyjmuje wektor o długości m)
- `strike` - strike opcji (przyjmuje wektor o długości m)
- `S0` - wartość początkowa aktywa bazowego
- `r` - stopa procentowa

4.3.2 Zwracany wynik funkcji

Funkcja zwraca wycenę dla opcji o zadanych parametrach (względnie, wektor wycen).

5 Obsługa programu

Przykład użycia opisanych funkcji przedstawiony jest w skrypcie `test.m` znajdującym się w głównym katalogu projektu. Korzysta on ze sztucznych danych wygenerowanych za pomocą funkcji `data_generator` z katalogu `data`. (przykład korzysta z danych generowanych sztucznie, ze względu na brak dostępu do powierzchni implied volatility w czasie pisania programu; zasadnicza logika nie zmienia się jednak dla danych rzeczywistych)

Aby uruchomić program należy wykonać następujące kroki:

1. Załadować zależności wykonując skrypt `requirements` (dodaje on jedynie poszczególne katalogi do ścieżki; jak już wspomniano projekt nie posiada żadnych zewnętrznych zależności).
2. Załadować dane z pliku (wygenerowane bądź rzeczywiste).
3. Stworzyć wektory `strikes`, `Ts` (nazwy mogą być inne) zawierające odpowiednio strike'i oraz czasy do wygaśnięcia odpowiadające kolumnom macierzy załadowanych danych.
4. Stworzyć wektory `ts`, `S0` zawierające odpowiednio kolejne punkty czasowe obserwacji i ceny aktywa bazowego w tych momentach, odpowiadające kolejnym wierszom macierzy załadowanych danych.
5. Stworzyć wektor początkowych wartości parametrów modelu Hestona `m_init`.
6. Ustalić stopę procentową `r`.
7. Wywołać funkcję `heston_calib_kalman` z odpowiednimi, stworzonymi wcześniej parametrami i odczytać jej wynik (patrz opis tej funkcji).

Literatura

[1] Lindström E., Ströjby J., Brodén M., *Sequential Calibration of Options*, 2008

[2] Moodley N., *The Heston Model: A Practical Approach with Matlab Code*, 2005