

Algorytmy statystyki praktycznej - projekt 2

Stanisław Wilczyński

12 kwietnia 2018

1 Wstęp

W tym projekcie zajmiemy się analizą rzutu ukośnego jako ukrytego modelu Markowa w kolejnych T jednostkowych odcinkach czasu. Naszym ukrytym stanem będą pozycja i prędkość przedmiotu rzuconego, a obserwowanymi wartościami jego pozycje (z pewnym losowym szumem). Bardziej formalnie:

$$\begin{aligned}X_k &= FX_{k-1} + B\mu + w_k, \text{ gdzie } w_k \sim \mathcal{N}(0, Q) \\Y_k &= HX_k + v_k, \text{ gdzie } v_k \sim \mathcal{N}(0, R) \\F &= \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\B &= \begin{pmatrix} 0 \\ -\frac{1}{2}\Delta t^2 \\ 0 \\ -\Delta t \end{pmatrix} \\H &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \\\mu &= 9.8 \\X_1 &\sim \mathcal{N}(\mu_0, \Sigma_0) \\\mu_0 &= \begin{pmatrix} 0 \\ 100 \\ 10 \\ 50 \end{pmatrix}\end{aligned}$$

Parametry Q i R ustalamy na odpowiednio $\frac{1}{1000}I_4$ oraz $\begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix}$. Będziemy próbować z naszych obserwacji wyznaczyć prawdziwe trajektorie X_1, \dots, X_T na podstawie Y_1, \dots, Y_T . Poza tym spróbujemy poradzić sobie z tym problemem zakładając brak znajomości Q i R .

2 Opis metod

2.1 Filtr Kalmana (FK)

Filtr Kalmana służy to wyznaczania ukrytych trajektorii (X_k) znając tylko wartości obserwacji poprzedzających dany ukryty stan (Y_1, \dots, Y_{k-1}). Metodę możemy opisać następującymi wzorami:

1. Inicjalizacja algorytmu

- $\hat{X}_{0|0} = \mu_0$
- $\Sigma_{0|0} = \Sigma_0$

2. Predykcja kolejnego stanu

- $\hat{X}_{k|k-1} = F\hat{X}_{k-1|k-1} + B\mu$

- $\Sigma_{k|k-1} = F\Sigma_{k-1|k-1}F^T + Q$

3. Filtracja

- $\hat{Y}_k = Y_k - H\hat{X}_{k|k-1}$
- $S_k = R + H\Sigma_{k|k-1}H^T$
- $K_k = \Sigma_{k|k-1}H^TS_k^{-1}$
- $\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k\hat{Y}_k$
- $\Sigma_{k|k} = \Sigma_{k|k-1} - K_kH\Sigma_{k|k-1}$

Kroki 2 i 3 wykonujemy w pętli dla $k = 1, 2, \dots, T$.

2.2 Wygładzanie Raucha–Tunga–Striebela (RTS)

Wygładzanie Raucha–Tunga–Striebela jest rozszerzeniem poprzedniej metody, uwzględniającym wszystkie obserwacje do estymowania ukrytych stanów. Tym razem zamiast przechodzić w przód, będziemy aktualizować wartości od końca, tzn. krok iteracyjny wykonujemy dla $k = T - 1, T - 2, \dots, 1$.

1. Inicjalizacja algorytmu - zastosowanie filtru Kalmana

2. Krok iteracyjny

- $L_k = \Sigma_{k|k}F^T\Sigma_{k+1|k}^{-1}$
- $\hat{X}_{k|T} = \hat{X}_{k|k} + L_k(\hat{X}_{k+1|T} - \hat{X}_{k+1|k})$
- $\Sigma_{k|T} = \Sigma_{k|k} + L_k(\Sigma_{k+1|T} - \Sigma_{k+1|k})L_k^T$

2.3 Algorytm EM

Algorytm EM służy do estymowania Q i R - w przypadku, gdy nie znamy macierzy kowariancji naszych szumów. Sama metoda wygląda następująco:

1. Inicjalizacja algorytmu - ustalamy początkowe Q i R - w naszym przypadku będą to odpowiednio I_4 i I_2
2. Krok iteracyjny - uruchamiamy procedurę estymującą ukryte stany (zarówno filtr Kalmana jak i wygładzanie RTS). Następnie obliczamy nowe wartości Q i R na podstawie aktualnych estymatorów z FK i RTS. Ze względu na brak wzorów na slajdach z wykładu [1] na estymatory największej wiarygodności Q w przypadku niezerowego μ , posłużymy się wzorami z [2]. Jedna iteracja algorytmu EM składa się z dwóch kroków:

(a) Krok Expectation

- $x_k := E(X_k|y_{1:T}) = \hat{X}_{k|T}$
- $x_k x_k^T := E(X_k X_k^T | y_{1:T}) = \hat{X}_{k|T} \hat{X}_{k|T}^T + \Sigma_{k|T}$

(b) Krok Maximization

- $Q = \frac{1}{T-1} \sum_{k=1}^{T-1} \left(\hat{X}_{k+1|T} - F\hat{X}_{k|T} - B\mu \right) \left(\hat{X}_{k+1|T} - F\hat{X}_{k|T} - B\mu \right)^T + F\Sigma_{k|T}F^T + \Sigma_{k+1|T} - \Sigma_{k+1|T}L_k^TF^T - FL_k\Sigma_{k+1|T}$
- $R = \frac{1}{T} \sum_{k=1}^T y_k y_k^T + Hx_k x_k^T H^T - y_k x_k^T H^T - Hx_k y_k^T$

Liczba iteracji algorytmu jest ustalana przez użytkownika.

3 Wyniki

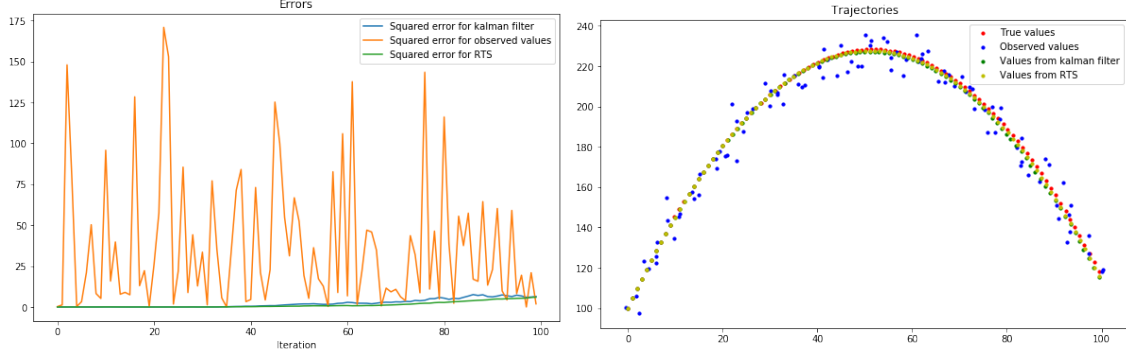
Działanie filtru Kalmana i wygładzania RTS będziemy badać biorąc pod uwagę następujące zmienne:

- T : 100 lub 1000
- Wybór punktu startowego dla algorytmów: dokładny ($\Sigma_0 = \Sigma_Z = 0$), losowy ($\Sigma_0 = \Sigma_R = \text{diag}(\mu_0) + 10I_4$) albo losowy o bardzo dużej macierzy kowariancji ($\Sigma_0 = \Sigma_X = 10^7 I_4$)
- Q i R : znane albo estymowane za pomocą algorytmu EM

3.1 T=100

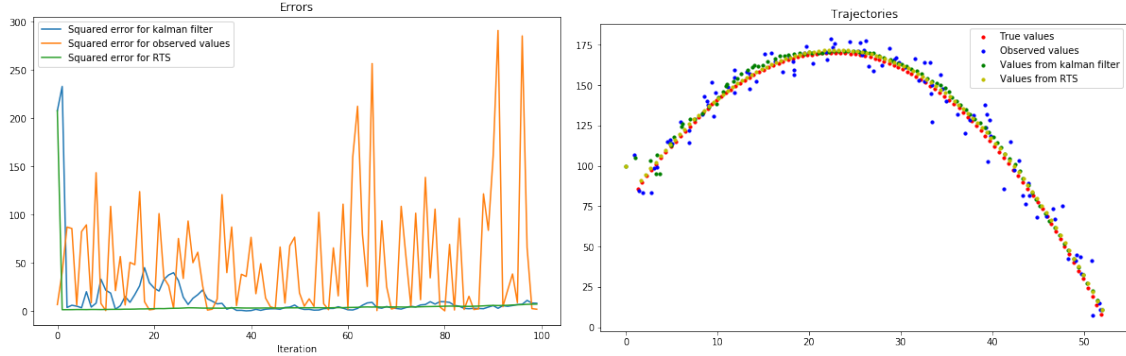
Poniższe dwa wykresy przedstawiają przypadek, w którym punkt startowy naszych trajektorii losujemy z $X_1 \sim \mathcal{N}(\mu_0, \Sigma_Z)$. Widzimy, że już od $k = 1$ estymowane \hat{X}_k są bardzo bliskie prawdziwemu X_k (lewy wykres). Trajektorie znalezione zarówno przez FK, jak i po zastosowaniu RTS prawie idealnie nachodzą na prawdziwe wartości (prawy wykres).

Rysunek 1: Odchylenia od trajektorii oraz trajektoria dla inicjalizacji dokładnej



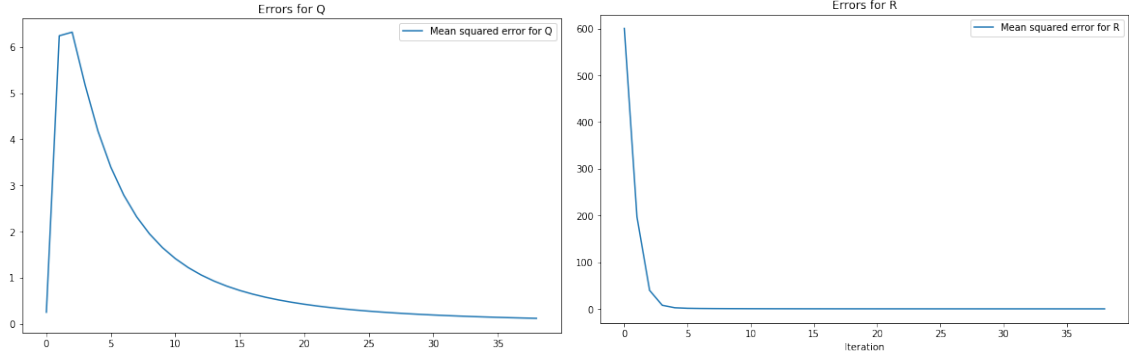
Drugim rozważanym przez nas przypadkiem będzie wylosowanie punktu startowego $X_1 \sim \mathcal{N}(0, \Sigma_R)$. Porównując dwa poniższe wykresy z dwoma powyższymi widzimy, że losowy punkt startowy nie jest dla naszych algorytmów dużym problemem. Już dla $k = 2$ (dla RTS) estymowane trajektorie zaczynają być bardzo blisko prawdziwych. Dla filtru Kalmana różnice widać zdecydowanie wyraźniej. Potwierdzają to nie tylko same wykresy, ale i wartości liczbowe: błąd średnio-kwadratowy dla obserwowanych punktów to 54.02, dla FK i RTS wynoszą 2.23, 1.30 oraz 12.63, 5.50 odpowiednio dla dokładnej i losowej inicjalizacji. W obu przypadkach widzimy, że RTS przybliża trajektorie zdecydowanie lepiej niż sam FK.

Rysunek 2: Odchylenia od trajektorii oraz trajektoria dla inicjalizacji losowej



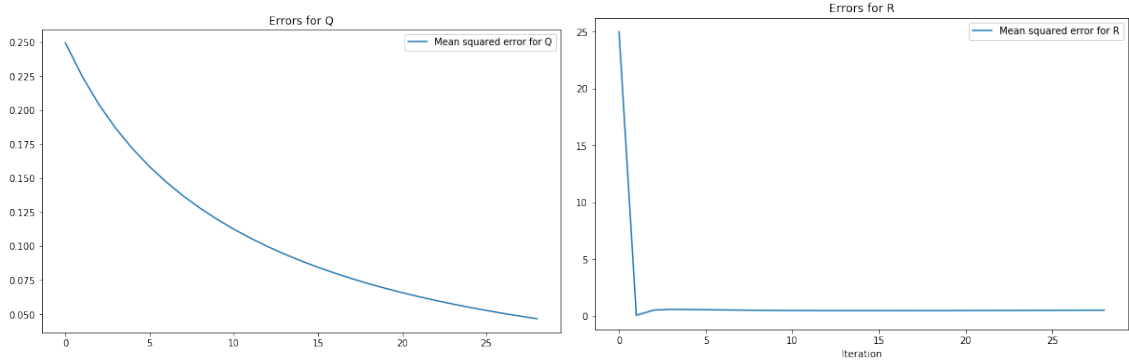
Następnie badaliśmy działanie algorytmu EM do estymowania Q i R w przypadku, gdy $\Sigma_0 = \Sigma_Z$. Otrzymaliśmy Q i R z błędami średnio-kwadratowymi (po elementach macierzy) wynoszącymi odpowiednio 0.12, 0.06. Porównaliśmy także otrzymane wyniki z algorytmem EM zaimplementowanym w bibliotece *pykalman* - różnice były minimalne dla Q na poziomie dwóch setnych, dla R trzech tysięcznych. Taki wynik może być spowodowany różnymi sposobami inicjalizacji w obu algorytmach (u nas są to odpowiednio I_4 i I_2). Na poniższym wykresie widzimy błędy średnio-kwadratowe dla estymowanych Q i R w kolejnych iteracjach algorytmu EM.

Rysunek 3: Błędy średnio-kwadratowe dla Q i R w algorytmie EM i $T = 100$



Taki wykres dla Q może się wydawać nieintuicyjny (ze względu na wzrost Q przy pierwszych iteracjach) ale jest to kwestia inicjalizacji naszych macierzy. Przy zmianie początkowego R na $\begin{pmatrix} 1 & 0 \\ 0 & 40 \end{pmatrix}$ otrzymujemy następujące wykresy:

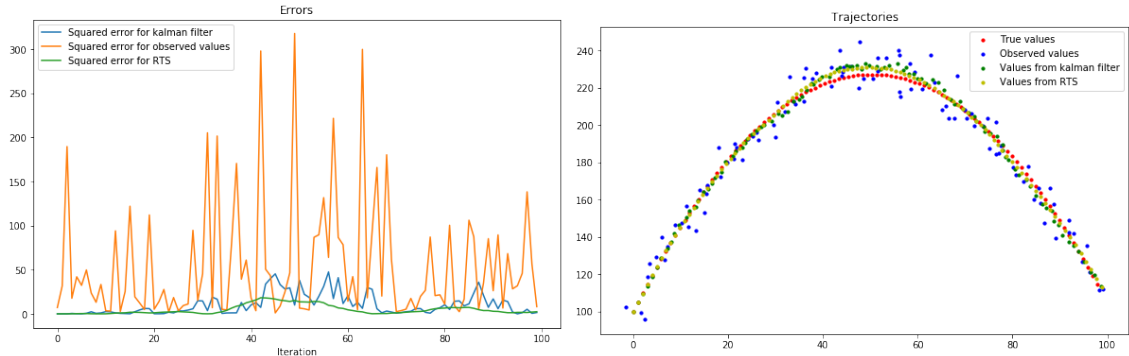
Rysunek 4: Błędy średnio-kwadratowe dla Q i R w algorytmie EM i $T = 100$ dla zmienionej inicjalizacji R



Jednak ze względu na chęć zachowania ogólności naszych metod zdecydowaliśmy się pozostać przy inicjalizacji Q i R odpowiedni I_4 i I_2 .

Teraz użyjemy filtru Kalmana i wygładzania RTS, które użyją estymowanych przez macierzy Q i R . Na poniższych wykresach widzimy, że tym razem obie metody nie zadziałały tak dobrze jak w przypadku inicjalizacji dokładnej. Wyraźne odchylenia od trajektorii pojawiają się nawet w okolicach 90 iteracji, a wartości liczbowe otrzymanych średnich błędów są porównywalne (tego samego rzędu) do tych dla z inicjalizacji losowej ($\Sigma_0 = \Sigma_R$).

Rysunek 5: Odchylenia od trajektorii oraz trajektoria dla inicjalizacji dokładnej po zastosowaniu EM

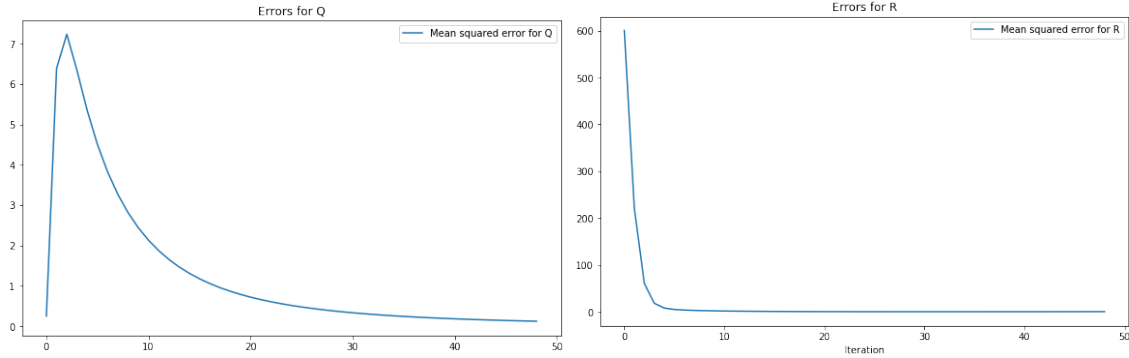


3.2 $T=1000$

Powyższe eksperymenty powtórzyliśmy również dla $T = 1000$. Tym razem wykresy odchyień od trajektorii oraz błędów ze względu na tak dużą liczbę punktów zostało pominięte. Ze względu na zwiększoną liczbę punktów, zmieniliśmy również wartość $\Sigma_R = \text{diag}(\mu_0)^2$. Otrzymane wartości liczbowe średnio-kwadratowych błędów dla FK i RTS wyniosły odpowiednio 0.81, 0.34 oraz 13.41, 5.83 odpowiednio dla inicjalizacji z $\Sigma_0 = \Sigma_Z$ i $\Sigma_0 = \Sigma_R$. W tym pierwszym wypadku możemy zaobserwować, więc zdecydowaną poprawę wartości naszych błędów, natomiast w drugim mimo znacznego zwiększenia kowariancji (ze względu na tak dużą liczbę iteracji) wyniki się niewiele pogorszyły.

Na poniższych wykresach podobnie jak poprzednio badamy skuteczność algorytmu EM do estymowania macierzy Q i R . Otrzymane błędy wyniosły odpowiednio 0.12, 0.15, więc nie widzimy znaczącej różnicy w porównaniu z $T = 100$, a i same wykresy wyglądają prawie identycznie.

Rysunek 6: Błędy średnio-kwadratowe dla Q i R w algorytmie EM i $T = 1000$

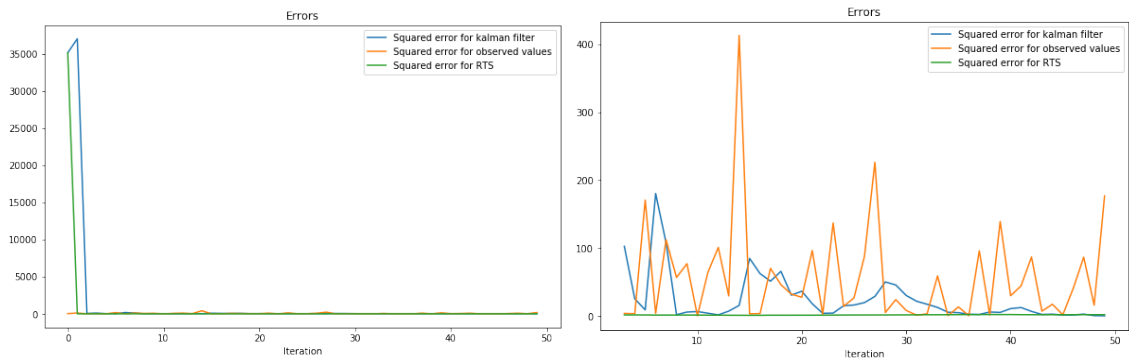


Jeśli chodzi o wartości błędów średnio-kwadratowych dla FK i RTS przy użyciu obliczonych powyżej estymatorów Q i R dostaliśmy 16.24, 6.98 dla odpowiednio filtra Kalmana i wygładzania RTS. Wyniki te podobnie jak w przypadku $T = 100$ są porównywalne (tego samego rzędu) z wartościami otrzymanymi przy inicjalizacji z użyciem $\Sigma_0 = \Sigma_R$.

3.3 Punkt startowy losowany z rozkładu o bardzo dużej kowariancji

Ostatnim eksperymentem było sprawdzenie jak sobie poradzą nasze metody przy ekstremalnie dużej macierzy kowariancji przy losowaniu X_1 wynoszącej $\Sigma_0 = \Sigma_X$ oraz niedużym $T = 50$ (zmniejszając w ten sposób efekt wygładzania). Na poniższych wykresach widzimy, że obie metody świetnie sobie z tym radzą. O ile w pierwszych 2 krokach różnice od właściwej trajektorii są ogromne, o tyle już po zaledwie trzech punktach nasze algorytmy już mają dużo mniejszy błąd przybliżenia niż same obserwacje. Dla filtra Kalmana widać jeszcze dość wyraźne oscylacje tego błędu, natomiast dla RTS jakość przybliżenia jest bardzo dobra (co również widać po trajektoriach), gdyż od trzeciego punktu błąd średnio-kwadratowy wynosi zaledwie 1.50.

Rysunek 7: Odchylenia od trajektorii dla $\Sigma_0 = \Sigma_X$



4 Podsumowanie

Na podstawie przeprowadzonych symulacji możemy stwierdzić, że wygładzanie Raucha–Tunga–Striebela wyraźnie poprawia wyniki działania filtru Kalmana. Poza tym testowaliśmy wpływ wyboru Σ_0 na działanie obu algorytmów. Możemy zauważyć, że zwiększenie współczynników w tej macierzy powoduje wyraźne pogorszenie jakości estymatorów. Następnie testowaliśmy działanie algorytmu EM do estymowania Q i R . Porównaliśmy działanie naszej metody z wersją z biblioteki *pykalman* i otrzymane rezultaty były bardzo podobne. Sama estymacja przy użyciu algorytmu EM nie była idealna. Po zastosowaniu FK i RTS z użyciem wyestymowanych Q i R otrzymaliśmy błędy średnio-kwadratowe dla trajektorii wyraźnie większe niż przy znanych Q i R . Kolejnym widocznym efektem jest to, że w raz ze wzrostem liczby obserwacji spada błąd średnio-kwadratowy dla obu FK i RTS, natomiast w przypadku algorytmu EM nie widać znaczącej poprawy. Ostatnim przeprowadzonym przez nas eksperymentem było sprawdzenie jak zachowują się FK i RTS jeśli za Σ_0 weźmiemy macierz o ekstremalnie dużych współczynnikach. Okazało się, że nasze algorytmy działają nawet w takim przypadku bardzo dobrze, gdyż po zaledwie trzech przeanalizowanych punktach nie było już widać wpływu takiej inicjalizacji.

Literatura

- [1] Błażej Miasojedow. Algorytmy statystyki praktycznej - slajdy. <https://www.mimuw.edu.pl/~bmia/algorytmy/algorytmy.pdf>, 2018. [Online; dostęp 03.04.2018].
- [2] Jared Wood. Learning for Robotics and Control - lecture notes. <https://inst.eecs.berkeley.edu/~cs294-40/fa08/scribes/lecture14.pdf>, 2008. [Online; dostęp 03.04.2018].