9.
Wersja rekurencyjna algorytmu najmniejszych kwadratów

wersja off-line (standardowa)

$$a_N = (X_N^T X_N)^{-1} X_N^T Y_N$$

wersja on-line (rekurencyjna)

$$a_N = f(a_{N-1}, x_N, y_N) = a_{N-1} + \delta(a_{N-1}, x_N, y_N)$$

odwracana macierz (w wersji off-line)

$$X_N^T X_N = \sum_{k=1}^{N} x_k x_k^T = \sum_{k=1}^{N-1} x_k x_k^T + x_N x_N^T = X_{N-1}^T X_{N-1} + x_N x_N^T$$

oznaczmy

$$P_N = (X_N^T X_N)^{-1}$$

$$cov(a_N) = P_N \sigma_z^2$$

możemy zapisać

$$\begin{aligned} a_N &= P_N X_N^T Y_N \\ a_{N-1} &= P_{N-1} X_{N-1}^T Y_{N-1} \end{aligned}$$

$$P_N = \left(P_{N-1}^{-1} + x_N x_N^T\right)^{-1}, \text{ gdzie } P_{N-1}^{-1} = X_{N-1}^T X_{N-1}$$

Lemat 1 *(o odwracaniu macierzy)*

$$\left(A + uu^T\right)^{-1} = A^{-1} - \frac{1}{1 + u^T A^{-1} u} A^{-1} u u^T A^{-1}$$

Przyjmując $A = P_{N-1}^{-1}$ i $u = x_N$ otrzymujemy

$$\begin{aligned} P_N &= P_{N-1} - \frac{1}{1 + x_N^T P_{N-1} x_N} P_{N-1} x_N x_N^T P_{N-1} = P_{N-1} - \varkappa_N P_{N-1} x_N x_N^T P_{N-1} \\ \text{gdzie } \varkappa_N &= \frac{1}{1 + x_N^T P_{N-1} x_N} \end{aligned}$$

$$\begin{aligned} a_N &= \left(P_{N-1} - \varkappa_N P_{N-1} x_N x_N^T P_{N-1}\right)\left(X_{N-1}^T Y_{N-1} + x_N y_N\right) = \\ &= P_{N-1} X_{N-1}^T Y_{N-1} + P_{N-1} x_N y_N - [\varkappa_N P_{N-1} x_N](x_N^T P_{N-1} X_{N-1}^T Y_{N-1} - x_N^T P_{N-1} x_N y_N) = \\ &= a_{N-1} + [\varkappa_N P_{N-1} x_N]\{\frac{1}{\varkappa_N} y_N - x_N^T a_{N-1} - x_N^T P_{N-1} x_N y_N\} \end{aligned}$$

wstawiając

$$\frac{1}{\varkappa_N} = 1 + x_N^T P_{N-1} x_N$$

otrzymujemy

$$\{...\} = y_N + x_N^T P_{N-1} x_N y_N - x_N^T a_{N-1} - x_N^T P_{N-1} x_N y_N = y_N - x_N^T a_{N-1}$$

wniosek

$$\begin{aligned} a_N &= a_{N-1} + \varkappa_N P_{N-1} x_N (y_N - x_N^T a_{N-1}) \\ P_N &= P_{N-1} - \varkappa_N P_{N-1} x_N x_N^T P_{N-1} \quad / \cdot x_N \end{aligned} \tag{1}$$

$$\begin{aligned} P_N x_N &= P_{N-1} x_N - \varkappa_N P_{N-1} x_N x_N^T P_{N-1} x_N = \varkappa_N P_{N-1} x_N \{\frac{1}{\varkappa_N} - x_N^T P_{N-1} x_N\} \\ \{...\} &= 1 \\ P_N x_N &= \varkappa_N P_{N-1} x_N \end{aligned}$$

wstawiamy do (1)

$$\begin{aligned} a_N &= a_{N-1} + P_N x_N (y_N - x_N^T a_{N-1}) \\ P_N &= P_{N-1} - \frac{P_{N-1} x_N x_N^T P_{N-1}}{1 + x_N^T P_{N-1} x_N} \end{aligned}$$

warunki początkowe

$$a_0 = 0,\ P_0 = diag[10^3 \div 10^5]$$

zaleta: algorytm pracuje bez odwracania macierzy