Hiukkassuotimet

Lasse Rintakumpu

24.helmikuuta 2021

Hiukkassuotimet

Hiukkassuotimet (*particle filters*, tunnetaan myös nimellä sekventiaaliset Monte Carlo -menetelmät, *sequential Monte Carlo*, SMC) ovat joukko 90-luvulta eteenpäin kehitettyjä menetelmiä, joiden avulla voidaan approksimoida epälineaarista Bayes-rekursiota / suodinongelmaa.

Menetelmissä käytetään otoksia (kutsutaan myös partikkeleiksi) esittämään jonkin stokastisen prosessin posteriorijakaumaa, kun vain osa havainnoista tunnetaan ja/tai havaintoihin liittyy kohinaa. Menetelmille on lukuisia sovellutuksia fysiikasta robotiikkaan.

Motivaatio: Suodinongelma

Hiukkassuotimen tavoitteena on estimoida "piilossa olevien" tilojen posteriorijakauma havaintojen perusteella. Tiedetään lisäksi miten havaitut muuttujat kytkeytyvät "piilossa oleviin" muuttujiin sekä osaamme sanoa jotain tilamuuttujien todennäköisyyksistä.

Eli tilanne on:

$$
\begin{array}{ccccccccl}
X_1 & \rightarrow & X_2 & \rightarrow & X_3 & \rightarrow & \dots & \text{piilossa olevat tilat} \\
\downarrow & & \downarrow & & \downarrow & & & \\
Y_1 & \rightarrow & Y_2 & \rightarrow & Y_3 & \rightarrow & \dots & \text{havainnot.}
\end{array}
$$

Motivaatio: Suodinongelma

$$\begin{array}{ccccccccl} X_1 & \rightarrow & X_2 & \rightarrow & X_3 & \rightarrow & \ldots & \text{piilossa olevat tilat} \\ \downarrow & & \downarrow & & \downarrow & & & \\ Y_1 & \rightarrow & Y_2 & \rightarrow & Y_3 & \rightarrow & \ldots & \text{havainnot.} \end{array}$$

Hiukkassuotimella estimoidaan *sekventiaalisesti* tilojen X_k arvot minä hyvänsä ajan hetkellä k, kun ainoastaan prosessi $Y_1, \ldots, Y_k$ tunnetaan. Estimaatit saadaan posteriorijakaumasta $p(x_k | y_1, y_2, \ldots, y_k)$, jolle siis muodostetaan Bayesilaisittain approksimaation havaintojen pohjalta.

Malli

Merkitään piilossa olevan prosessin tilaa ajanhetkellä k x_k ja havaittua prosessia y_k. Aika-avaruus on diskreetti. Hiukkassuodinta varten määritellään prosesseille mallit:

$$
\begin{aligned}
x_{k+1} &\sim p(x_{k+1}|x_k) \\
y_k &\sim p(y_k|x_k)
\end{aligned}
$$

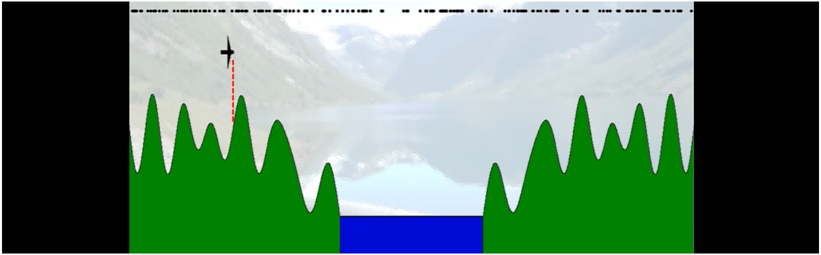
joiden pohjalta halutaan laskea posteriorijakauma $p(x_k|y_{1:k})$.

Motivaatioesimerkki: Paikantaminen

Esimerkissä haluamme tietää lentokoneen sijannin kaksiulotteisessa maailmassa, kun tiedämme

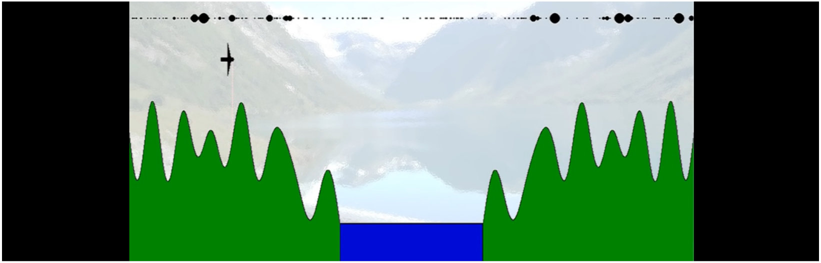
- lentokorkeuden,
- etäisyyden maanpinnasta
- ja käytössämme on maastokartta (mutta emme tiedä missä olemme).
- Kuvat varastettu animaatiosta **Particle Filter Explained without Equations**, https://www.youtube.com/watch?v=aUkBa1zMKv4.

Motivaatioesimerkki: Paikantaminen



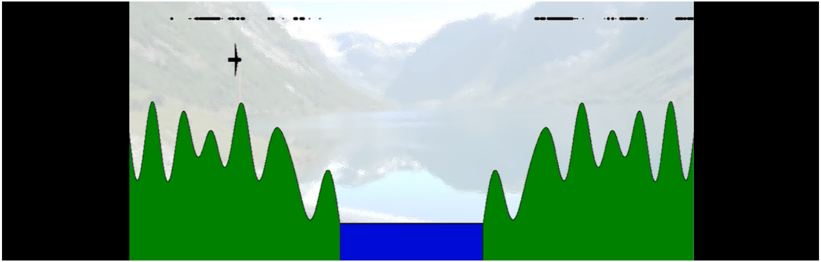
Kuva 1: Alustus

Motivaatioesimerkki: Paikantaminen



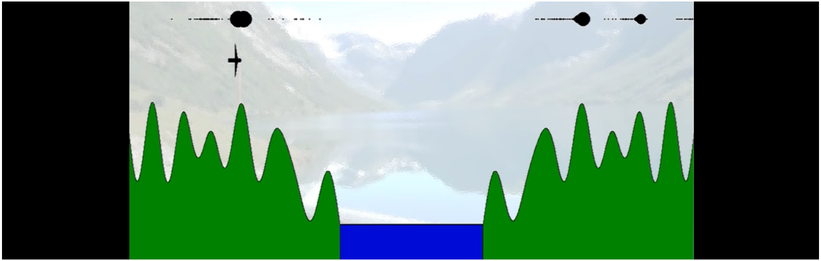
Kuva 2: Mittaus ja painojen päivitys

Motivaatioesimerkki: Paikantaminen



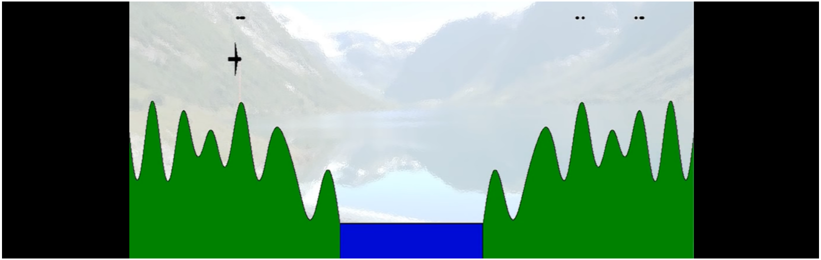
Kuva 3: Uudellenotanta ja ajan päivitys

Motivaatioesimerkki: Paikantaminen



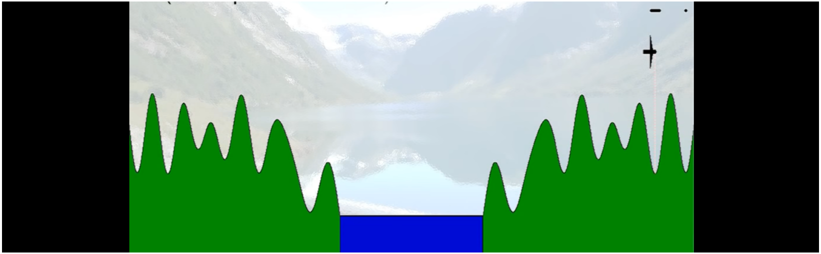
Kuva 4: Mittaus ja painojen päivitys

Motivaatioesimerkki: Paikantaminen



Kuva 5: Uudelleenotanta ja ajan päivitys

Motivaatioesimerkki: Paikantaminen



Kuva 6: Ja niin edelleen

Bayesilainen suodin

Bayesilainen ratkaisu posteriorijakaumalle saadaan seuraavalla rekursiolla (käydään läpi jokaiselle ajanhetkelle $k = 1, \ldots, t$). Lasketaan ensin

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$$

joka saadaan suoraan Bayesin kaavasta $P(A|B) = P(B|A)P(A)/P(B)$. Tämä vastaa lentokone-esimerkissä mittaustulosten päivittämistä.

Bayesilainen suodin

Normalisointivakio lasketaan integraalina

$$p(y_k|y_{1:k-1}) = \int_{\mathbb{R}^{n_x}} p(y_k|x_k)p(x_k|y_{1:k-1})\,dx_k$$

joka saadaan kokonaistodennäköisyyskaavasta $P(A) = \mathbb{E}[P(A|X)] = \int_{-\infty}^{\infty} P(A|X = x)f_X(x)\,dx$. Merkintä R^{n_x} vastaa tässä piilosa olevien muuttujien dimensiota n.

Bayesilainen suodin

Lopuksi lasketaan päivitysaskel ajalle, joka saadaan edelleen kokonaistodennäköisyydellä

$$p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^{n_x}} p(x_{k+1}|x_k)p(x_k|y_{1:k})\,dx_k\,.$$

Rekursion avulla voidaan siis laskea $p(x_k|y_{1:k})$ käymällä rekursio läpi k kertaa.

Hiukassuodin-algoritmi

Ennen algoritmin suorittamista valitaan ehdotusjakauma $q(x_{k+1}|x_{1:k}, y_{k+1})$, uudelleenotantamenetelmä sekä partikkelien määrä N. Ehdotusjakauman valintaa ei käsitellä tässä, mutta valinta voidaan tehdä optimaalisesti. Yksinkertaisin valinta on $q(x_{k+1}|x_{1:k}, y_{k+1}) = p(x_{k+1}|x_{1:k})$.

Hiukassuodin-algoritmi

Alla käytetään notaatiota x_k^i, joka tarkoittaa, että tila x_k käy ajanhetkellä k gridin pisteessä x^i. Notaatiota tarvitaan, koska hiukkassuotimessa läpikäytävä gridi muuttuu ajan funktiona.

Alustus: Generoidaan $x_1^i \sim p_{x_0}$ missä $i = 1, \ldots, N$ ja jokainen partikkeli saa saman painon $w_{1|0}^i = 1/N$.

Hiukassuodin-algoritmi

Jokaiselle ajanhetkelle $k = 1, 2, \ldots, t$ toistetaan seuraava laskenta.

Vaihe 1: Päivitetään havainnot. Jokaiselle partikkelille $i = 1, 2, \ldots, N$ lasketaan painot

$$w^i_{k|k} = \frac{1}{c_k} w^i_{k|k-1} p(y_k | x^i_k)$$

missä normalisointipaino on

$$c_k = \sum_{i=1}^{N} w^i_{k|k-1} p(y_k | x^i_k).$$

Hiukassuodin-algoritmi

Vaihe 2: Estimoidaan p. Lasketaan tiheydelle approksimaatio

$$\hat{p}(x_{1:k}|y_{1:k}) = \sum_{i=1}^{N} w_{k|k}^{i} \delta(x_{1:k} - x_{1:k}^{i})$$

missä $\delta(x)$ on Diracin deltafunktio.

Hiukassuodin-algoritmi

Vaihe 3: Valinnainen uudelleenotanta. Otetaan uudet N otosta palauttaen joukosta $\{x^i_{1:k}\}_{i=1}^N$ missä otoksen i todennäköisyys on $w^i_{k|k}$.

Uudelleenotantaa tarvitaan, koska ilman sitä painot alkavat keskittyä muutaman ajanhetken jälkeen tietyille partikkeleille. Uudelleenotantaa ei kuitenkaan kannata aloittaa liian aikaisin, koska se lisää satunnaisotannan epävarmuutta.

Hiukassuodin-algoritmi

Vaihe 4: Aikapäivitys, jos $k < t$. Luodaan ennusteet partikkeleille ehdotusjakaumasta

$$x_{k+1}^i \sim q(x_{k+1} | x_k^i, y_{k+1})$$

ja päivitetään myös partikkelien painot tärkeytysotannalla sen mukaan kuinka todennäköisiä partikkelien ennusteet ovat

$$w_{k+1|k}^i = w_{k|k}^i \frac{p(x_{k+1}^i | x_k^i)}{q(x_{k+1}^i | x_k^i, y_{k+1})}.$$

Palataan takaisin **vaiheeseen 1**. Jatketaan kunnes $k = t$. Kun $N \to \infty$ algoritmille pätee $\hat{p}(x_{1:k} | y_{1:k}) \xrightarrow{a.s.} p(x_{1:k} | y_{1:k})$.

Marginaalijakauma

Edellä kuvattu algoritmi tuottaa approksimaation koko prosessin posteriorijakaumalle $p(x_{1:k}|y_{1:k})$. Jos halutaan tietää ainoastaan posteriorijakauman $p(x_k|y_{1:k})$ estimaatti, voidaan käyttää ainoastaan viimeisestä tilasta x_k^i laskettua estimaattia

$$\hat{p}(x_k|y_{1:k}) = \sum_{i=1}^{N} w_{k|k}^i \delta(x_k - x_k^i).$$

Marginaalijakauma

Toinen vaihtoehto on käyttää laskennassa tärkeytyspainoa

$$w^i_{k+1|k} = \frac{\sum_{j=1}^{N} w^j_{k|k} p(x^i_{k+1}|x^j_k)}{q(x^i_{k+1}|x^i_k, y_{k+1})}.$$

yllä esitetyn sijaan. Tällöin jokaisessa aikapäivitysaskeleessa lasketaan painot kaikkien mahdollisten tila-aika-avaruuspolkujen yli. Samoin kuin uudelleenotanta tämä pienentää painojen varianssia.

Menetelmä kuitenkin lisää algoritmin aikakompleksisuutta $\mathcal{O}(N) \to \mathcal{O}(N^2)$, joten isoilla arvoilla N pitää käyttää jotakin toista versiota algoritmista. Esimerkiksi $\mathcal{O}(N\log(N))$ versio tästä marginaalihiukkassuotimesta on olemassa.

Kirjallisuutta

- Dahlin & Schön (2019): **Getting Started with Particle Metropolis-Hastings for Inference in Nonlinear Dynamical Models**, https://arxiv.org/pdf/1511.01707.pdf,
- Gordon, Salmond & Smith (1993): **Novel approach to nonlinear/non-Gaussian Bayesian state estimation**, http://www.irisa.fr/aspi/legland/ref/gordon93a.pdf,
- Gustafsson (2010): **Particle Filter Theory and Practice with Positioning Applications**, https://ieeexplore.ieee.org/document/5546308,
- Wikipedia: **Particle filter**, https://en.wikipedia.org/wiki/Particle_filter.