

SMC-menetelmät

Laskennallinen tilastotiede - Harjoitustyö

Lasse Rintakumpu

28 April, 2021

Sisällys

| | |
|---|----------|
| 1 Johdanto | 1 |
| 1.1 Suodinongelma | 1 |
| 1.2 Historiaa | 2 |
| 2 Bayesilainen suodin | 2 |
| 3 Hiukassuodin-algoritmi | 3 |
| 3.1 Parametrien valinta | 3 |
| 3.1.1 Otoksoon N valinta | 3 |
| 3.1.2 Uudelleenotantamenetelmän valinta | 3 |
| 3.1.3 Ehdotusjakauman valinta | 3 |
| 3.2 Algoritmi | 4 |
| 3.3 Asymptotiikkaa | 4 |
| 3.4 Marginaalijakauma | 4 |
| 3.5 Aikakompleksisuus | 5 |
| 4 Paikannusesimerkki | 5 |
| 5 Lopuksi | 5 |
| 6 Lähteet | 5 |

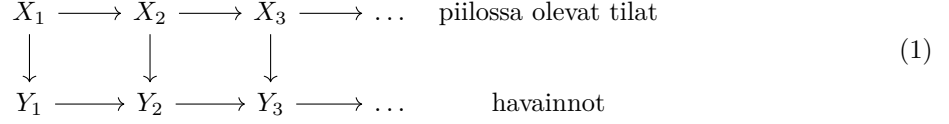
1 Johdanto

SMC-menetelmät (Sequential Monte Carlo -menetelmät, tunnetaan myös nimellä hiukassuotimet) ovat joukko 90-luvulta eteenpäin kehitettyjä Monte Carlo -algoritmeja, joiden avulla voidaan ratkaista ns. suodinongelma, kun ongelma on epälineaarinen ja/tai ongelmaan liittyvä kohina ei noudata normaalijakaumaa. Tämän tutkielman tavoitteena on esittää pääpiirteittäin SMC-menetelmien teoria sekä joitakin menetelmäperheeseen kuuluvia algoritmeja. Tutkielman esitykset seuraavat erityisesti Simo Särkän kirjaa *Bayesian Filtering and Smoothing* (2013) sekä Fredrik Gustafssonin artikkelia "Particle Filter Theory and Practice with Positioning Applications" (2010). SMC-menetelmille on lukuisia sovellutuksia esimerkiksi Bayesilaisessa tilastotieteessä, fysiikassa ja robotiikassa. Tämän tutkielman lopussa tarkastellaan hiukassuotimen käyttöä paikannussovelluksessa.

1.1 Suodinongelma

Stokastisten prosessien teoriassa suodinongelmaksi kutsutaan tilannetta, jossa halutaan muodostaa paras mahdollinen estimaatti jonkin järjestelmän tilan arvoille, kun ainoastaan osa tiloista voidaan havaita ja/tai havaintoihin liittyy kohinaa. Tavoitteena on siis laskea jonkin Markov-prosessin posteriorijakauma kyseisten

havaintojen perusteella. Tässä tutkielmassa keskitytään erityisesti ns. Markovin piilomallin posteriorijakauman Bayesilaiseen ratkaisuun, kun itse ongelma on EPÄLINEAARINEN. Ongelmaa havainnollistaa kaavio (1).



Ongelmassa tiedämme, miten havaitut muuttujat kytkeytyvät “piilossa oleviin” muuttujiin sekä osaamme sanoa jotain tilamuuttujien todennäköisyyksistä. Oletamme lisäksi, että piilossa olevat tilat muodostavat Markovin ketjun. Kun aika-avaruus on diskreetti ja merkitsemme piilossa olevan prosessin tilaa ajanhetkellä k x_k ja havaittua prosessia y_k , meillä on siis olemassa mallit

$$x_{k+1} \sim p(x_{k+1}|x_k) \tag{2}$$

$$y_k \sim p(y_k|x_k). \tag{3}$$

Näin määritettyjen mallien avulla SMC-menetelmät estimoivat sekventiaalisesti tilojen X_k arvot minä hyvänsä ajan hetkellä k , kun ainoastaan prosessi Y_1, \dots, Y_k tunnetaan. Estimaatit saadaan posteriorijakaumasta $p(x_k|y_1, y_2, \dots, y_k)$, jonka approksimaatio muodostetaan Bayesilaisittain havaintojen pohjalta. Kuten mainittua ei *** tarvitse OLLA LINEAARINEN. SMC-menetelmissä stokastisen prosessin posteriorijakauman esittämiseen käytettyjä otoksia kutsutaan myös partikkeleiksi.

1.2 Historiaa

Lineaarisen suodinongelman ratkaisun

Epälineaarille ei-Gaussilaiselle mallille ... Muita vaihtoehtoja ovat EKF, UKF, QKF.

Ensimmäisen epälineaarisen suodinongelman Bayesilaisen/MC-ratkaisun esittivät Gordon, Salmond ja Smith artikkelissaan “Novel approach to nonlinear/non-Gaussian Bayesian state estimation” (1993). Gordonin, Salmondin ja Smithin ratkaisu eroaa notaatioltaan hieman tässä tutkielmassa esitetystä, mutta on olennaisesti sama. Suurin ero tämän tutkielman SMC-algoritmin sekä alkuperäisen SMC-algoritmin välillä on XXX. Artikkelissa ratkaisu kulki nimellä “bootstrap filter”, saapasremmisuodin. MIKSI Termiä hiukkassuodin käytti ensimmäisen kerran Del Moral artikkelissa “Nonlinear Filtering: Interacting Particle Resolution” (1996), SMC-menetelmät termiä Liu ja Chen artikkelissa “Sequential Monte Carlo Methods for Dynamic Systems” (1998). Tässä tutkielmassa pyritään korostamaan suotimien yhteyttä Monte Carlo -algoritmeihin ja käytetään siksi termiä SMC-menetelmät.

2 Bayesilainen suodin

Ennen SMC-algoritmia käydään läpi algoritmeissa käytetty yleinen Bayesilainen posteriorijakauman laskenta. Esitys noudattaa Fredrik Gustafssonin artikkelia “Particle Filter Theory and Practice with Positioning Applications” (2010). Bayesilainen ratkaisu tilavektorin posteriorijakaumalle $p(x_k|y_{1:k})$ saadaan seuraavalla rekursiolla (käydään läpi jokaiselle ajanhetkelle $k = 1, \dots, t$). Lasketaan ensin

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}, \tag{4}$$

joka saadaan suoraan Bayesin kaavasta $P(A|B) = P(B|A)P(A)/P(B)$. Normalisointivakio lasketaan integraalina

$$p(y_k|y_{1:k-1}) = \int_{\mathbb{R}^{n_x}} p(y_k|x_k)p(x_k|y_{1:k-1}) dx_k, \tag{5}$$

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

Lopuksi lasketaan päivitysaskel ajalle, joka saadaan edelleen kokonaistodennäköisyydellä

$$p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^{n_x}} p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k. \quad (6)$$

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

3 Hiukassuodin-algoritmi

Tässä luvussa esitetään algoritmi, joka hyödyntää luvussa 2. kuvattua Bayesilaista suodinta epälineaarisen suodinongelman ratkaisemiseen. Ennen varsinaista algoritmia käsitellään algoritmiin liittyvän uudelleenotantamenetelmän, partikkelien määrän / otoskoon ja ehdotusjakauman valinta. Lopuksi esiteetään algoritmin konvergenssia, marginaalijakaumaa sekä aikakompleksisuutta koskevia tuloksia. Esitetty algoritmi perustuu Fredrik Gustafssonin artikkeliin “Particle Filter Theory and Practice with Positioning Applications” (2010).

3.1 Parametrien valinta

Ennen algoritmin suorittamista valitaan ehdotusjakauma $q(x_{k+1}|x_{1:k}, y_{k+1})$, uudelleenotantamenetelmä sekä partikkelien määrä N . LYHYT KUVAAUS kaikista.

3.1.1 Otoksoon N valinta

3.1.2 Uudelleenotantamenetelmän valinta

Ilman uudelleenotantaa on todennäköistä, että algoritmi alkaa kärsiä otosten ehtymisestä. Toisin sanoen kaikki painot alkavat keskittyä vain muutamalle havainnolle. Uudelleenotanta tarjoaa osittaisen ratkaisun tähän ongelmaan, mutta hävittää samalla informaatiota ja siten lisää satunnaisotantaan liittyvää epävarmuutta. Yleisesti ottaen kannattaa uudelleenotanta aloittaa vasta siinä vaiheessa algoritmin suorittamista, kun siitä on otosten ehtymisen kannalta hyötyä.

Alla esitetty algoritmi käyttää SIR-uudelleenotantaa (sampling importance resampling), jossa uudelleenotanta suoritetaan jokaisella algoritmin läpikäynnillä. Vaihtoehtona on hyödyntää tärkeytysotantaa ja suorittaa uudelleenotanta ainoastaan, kun otoskoon ehtymisen mittarina käytettävä efektiivinen otoskoko painuu jonkin kynnyksarvon alapuolelle. Tätä kutsutaan SIS-uudelleenotannaksi (sampling importance sampling).

Efektiivinen otoskoko saadaan laskettua variaatiokertoimesta c_ν lasketaan kaavalla

$$N_{eff} = \frac{N}{1 + c_\nu^2(w_{k|k}^i)} = \frac{N}{1 + \frac{\text{Var}(w_{k|k}^i)}{(\mathbb{E}[w_{k|k}^i])^2}} = \frac{N}{1 + N^2 \text{Var}(w_{k|k}^i)}. \quad (7)$$

Tästä saadaan efektiiviselle otoskoolle laskennallinen approksimaatio

$$\hat{N}_{eff} = \frac{1}{\sum_i (w_{k|k}^i)^2}. \quad (8)$$

Approksimaatiolle pätee JOSSA yläraja. Tästä saadaan määriteltyä algoritmille SIS-uudelleenotantaehto $\hat{N}_{eff} < N_{th}$. Gustafsson (2010) esittää uudelleenotannan kynnyksarvoksi esimerkiksi $\hat{N}_{th} = 2N/3$.

3.1.3 Ehdotusjakauman valinta

Yksinkertaisin valinta ehdotusjakaumalle on $q(x_{k+1}|x_{1:k}, y_{k+1}) = p(x_{k+1}|x_{1:k})$. Ehdotusjakauman valintaa ei käsitellä tässä, mutta valinta voidaan tehdä optimaalisesti.

3.2 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 $i = 1, \dots, N$ ja jokainen partikkeli saa saman painon $w_{1|0}^i = 1/N$. Jokaiselle ajanhetkelle $k = 1, 2, \dots, t$ toistetaan seuraava laskenta.

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

$$w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k | x_k^i),$$

missä normalisointipaino on

$$c_k = \sum_{i=1}^N w_{k|k-1}^i p(y_k | x_k^i).$$

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.

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

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

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 $k = t$.

3.3 Asymptotiikkaa

Kun $N \rightarrow \infty$ algoritmille pätee $\hat{p}(x_{1:k} | y_{1:k}) \xrightarrow{a.s.} p(x_{1:k} | y_{1:k})$.

3.4 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

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

Toinen vaihtoehto on käyttää laskennassa tärkeytyspainoa

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

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

3.5 Aikakompleksisuus

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

4 Paikannusesimerkki

Esimerkissä analysoidaan ja vertaillaan algoritmin eri versioiden suorituskykyä. Esimerkkidatana käytetään Walkbasen XR-2 MITÄ.

5 Lopuksi

6 Lähteet

- 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>.
- Särkkä (2013): **Bayesian Filtering and Smoothing**. Cambridge University Press. https://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf.