SMC-menetelmät

Laskennallinen tilastotiede - Harjoitustyö

Lasse Rintakumpu

28 April, 2021

Sisällys

1 Johdanto

SMC-menetelmät (Sequential Monte Carlo -mentelmät, tunnetaan myös nimellä hiukkassuotimet) 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 hiukkassuotimen 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).

$$\begin{array}{ccccccc} X_1 & \longrightarrow & X_2 & \longrightarrow & X_3 & \longrightarrow & \dots \quad \text{piilossa olevat tilat} \\ \downarrow & & \downarrow & & \downarrow & & \\ Y_1 & \longrightarrow & Y_2 & \longrightarrow & Y_3 & \longrightarrow & \dots \quad \text{havainnot} \end{array} \tag{1}$$

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.

SUODINongelmaa lähellä on myös ns. s (*smoothing problem*), jossa ollaan kiinnostuneita. Tämä tutkielma keskittyy yksin suodinongelman ratkaisemiseen.

1.2 Historiaa

Lineaarisen suodinongelman ratkaisun ... .

Epälineaariselle 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. Poikkeuksen tähän tekee varsinainen esitetty algoritmi, jota kutsutaan tutkielmassa hiukkassuodin-algoritmiksi.

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 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 piilossa olevien muuttujien dimensiota n.

Lopuksi lasketaan päivitysaskel ajalle, joka saadaan edelleen kokonaistodennäköisyydellä

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

Rekursion 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. Ehdotusjakauman ja uudelleenotantamenetelmän valinnassa tärkeimpänä päämääränä on välttää otosten ehtymistä, kun taas partikkelien määrä säätelee kompromissia algoritmin suorituskyvyn ja tarkkuuden välillä.

3.1.1 Otoskoon N valinta

Yleispätevää sääntöä otoskoon/partikkelien lukumäärän N valinnalle on vaikeaa antaa, sillä vaadittava estimointitarkkuus riippuu usein käsillä olevasta ongelmasta. Gordon &al. (1993) esittävät kuitenkin kolme tekijää, jotka vaikuttavat partikkelien lukumäärän valintaan

a. tila-avaruuden ulottuvuuksien lukumäärä n_x,
b. tyypillinen päällekäisyys priorin ja uskottavuuden välillä
c. sekä tarvittava aika-askeleiden lukumäärä.

Ensimmäisen tekijän vaikutus on selvä. Mitä useammassa ulottuvuudessa otantaa tarvitsee tehdä, sen korkeammaksi on N asetettava, jotta jokainen ulottuvuus pystytään kattamaan. Tekijät (b) ja (c) puolestaan seuraavat uudelleenotannasta. Jos se osa tila-avaruutta, jossa uskottavuus $p(y_k|x_k)$ saa merkittäviä arvoja on pieni verrattuna siihen osaan, jossa priorijakauma $p(x_k|y_{1:k-1})$ saa merkittäviä arvoja, suuri osa partikkeleista saa pieniä painoja eivätkä näin valikoidu uudelleenotantaan.

Yleisesti ottaen N kannattaa asettaa sellaiseksi, että se paitsi tuottaa riittävän tarkan estimaatin, on se käytettävissä olevan laskentatehon sekä vaadittavan laskentanopeuden kannalta järkevää. Tähän palataan tutkielman lopuksi empiirisessä paikannusesimerkissä.

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.

Jos alla esitetyssä algoritmissa uudelleenotanta suoritetaan jokaisella algoritmin läpikäynnillä on kyseessä ns. SIR-algoritmi (sampling importance resampling). Vaihtoehtona on hyödyntää tärkeytysotantaa ja suorittaa uudelleenotanta ainoastaan, kun otoskoon ehtymisen mittarina käytettävä efektiivinen otoskoko painuu jonkin kynnysarvon alapuolelle. Tätä kutsutaan SIS-algoritmiksi (sampling importance sampling).

Efektiivinen otoskoko saadaan laskettua variaatiokertoimesta c_ν lasketaan kaavalla

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

Näin laskettu efektiivinen otoskoko maksimoituu ($N_{eff} = N$), kun kaikille painoille pätee $w_{k|k}^i = 1/N$ ja minimoituu ($N_{eff} = 1$), kun $w_{k|k}^i = 1$ todennäköisyydellä $1/N$ ja $w_{k|k}^i = 0$ todennäköisyydellä $(N-1)/N$. Tästä saadaan effektiiviselle otoskoolle laskennallinen approksimaatio

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

Sekä määritelmälle (7) että (8) pätee $1 \leq \hat{N}_{eff} \leq N$. Yläraja saavutetaan, kun jokaisen partikkelin paino on sama. Alarajalle puolestaan päädytään, kun kaikki paino päätyy yksittäiselle partikkelille. Tästä saadaan määriteltyä algoritmille SIS-uudelleenotantaehto $\hat{N}_{eff} < N_{th}$. Gustafsson (2010) esittää uudelleenotannan kynnysarvoksi esimerkiksi $\hat{N}_{th} = 2N/3$.

Uudelleenotanta ei muuta approksimoitavan jakauma p odotusarvoa, mutta se lisää jakauman Monte Carlo -varianssia. On kuitenkin olemassa uudelleenotantamenetelmiä, jotka pyrkivät minimoimaan tämän varianssin lisäyksen. Varianssitarkastelu jätetään tämän tutkielman ulkopuolelle.

3.1.3 Ehdotusjakauman valinta

Yksinkertaisin valinta ehdotusjakaumalle on $q(x_{1:k}|y_{1:k})$ eli toisin sanoen jokaisella algoritmin suorituskerralla käydään läpi koko polku $1:k$. Tämä ei kuitenkaan ole tarkoituksenmukaista, erityisesti jos kyseessä on reaaliaikainen sovellutus. Kirjoitetaan tämä ehdotusjakauma nyt muodossa

$$q(x_{1:k}|y_{1:k}) = q(x_k|x_{1:k-1}, y_{1:k}) q(x_{1:k-1}|y_{1:k}). \tag{9}$$

Jos yhtälöstä (9) poimitaan ehdotusjakaumaksi ainoastaan termi $q(x_k|x_{1:k-1}, y_{1:k})$ saadaan tämän avulla muodostettua hyvä approksimaatio arvoille x_k. Tämä on suodinongelman kannalta riittävää, koska olemme kiinnostuneita ainoastaan posteriorijakaumasta ajanhetkellä k (tasoitusongelmassa tarvitsisimme koko polun $x_{1:k}$). Alla tarkastellaan edelleen Gustafssonia (2010) seuraten joitakin ehdotusjakauman valintatapoja.

3.2 Algoritmi

Alla esitetään SMC-algoritmi Bayesilaisen, epälineaarisen suodinongelman ratkaisemiseksi. Kun uudelleenotanta tehdään jokaisella algoritmin iteraatiolla, on kyseessä SIR-algoritmi. Kun uudelleenotanta tehdään esimerkiksi edellisessä alaluvussa kuvatun kynnysarvoehdon $\hat{N}_{eff} < N_{th}$ täyttessä, on kyseessä SIS-algoritmi. Algoritmin kuvauksessa käytetään notaatiota x_k^i, joka tarkoittaa, että tila x_k käy ajanhetkellä k gridin pisteessä x^i. Notaatiota tarvitaan, koska SMC-algoritmin läpikäymä gridi muuttuu ajan funktiona.

3.3 Konvergenssituloksia

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

Algorithm 1: Hiukassuodin

Data: Generoidaan $x_1^i \sim p_{x_0}$ missä $i = 1, \ldots, N$ ja jokainen partikkeli saa saman painon $w_{1|0}^i = 1/N$.
Result: Posteriorijakauma x_{k+1}^i.
begin
  for $k = 1, 2, \ldots, t$ **do**
    for $i = 1, 2, \ldots, N$ **do**
      begin
        Päivitetään painot. $w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k|x_k^i)$,
        missä normalisointipaino on $c_k = \sum_{i=1}^N w_{k|k-1}^i p(y_k|x_k^i)$.
      begin
        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.
      begin
        Lasketaan efektiivinen otoskoko $\hat{N}_{eff} = \frac{1}{\sum_i (w_{k|k}^i)^2}$.
      if $\hat{N}_{eff} < N_{th}$ **then**
        Otetaan uudet N otosta palauttaen joukosta $\{x_{1:k}^i\}_{i=1}^N$, missä otoksen i todennäköisyys on $w_{k|k}^i$.
      if $\lessgtr t$ **then**
        Aikapäivitys.
        Luodaan ennusteet partikkeleille ehdotusjakaumasta $x_{k+1}^i \sim q(x_{k+1}|x_k^i, y_{k+1})$,
        päivitetään partikkelien painot tärkeytysotannalla sen mukaan kuinka todennäköisiä
        partikkelien ennusteet ovat $w_{k+1|k}^i = w_{k|k}^i \frac{p(x_{k+1}^i|x_k^i)}{q(x_{k+1}^i|x_k^i, y_{k+1})}$.

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. The sensor is being built around the BLE5.1 angle of arrival (AoA) specifications, enabling it to produce sub meter positioning results. Seitsemän sensoria sekä 25 paikannustagista koostuva testilaite. Riippuen päällä olevien paikannustagien lukumäärästä, tuottaa n. havainto sekunnissa. Jokainen havainto koostuu seuraavista muuttujista:

5 Lopuksi

Tässä tutkielmassa on esitetty pääpiirteittäin ... ja lisäksi tarkasteltu miten eri valinnat vaikuttavat algoritmin suorituskykyyn yksinkertaisen mutta XXX esimerkin avulla. Ei oteta kantaa esimerkiksi JATKOKYSYMYKSIÄ mahdollisuus laajentaa.

6 Lähteet

- Chen & Liu (1998): **Sequential Monte Carlo Methods for Dynamic Systems**. http://stat.rutgers.edu/home/rongchen/publications/98JASA_SMC.pdf
- Dahlin & Schön (2019): **Getting Started with Particle Metropolis-Hastings for Inference in Nonlinear Dynamical Models**. https://arxiv.org/pdf/1511.01707.pdf.
- Del Moral (1996): **Nonlinear Filtering: Interacting Particle Resolution**. https://people.bordeaux.inria.fr/pierre.delmoral/delmoral96nonlinear.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.