

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i: STK4050/9050 — Statistiske
simuleringer og numeriske
beregninger.

Eksamensdag: Tirsdag 6. desember 2005.

Tid for eksamen: 14.30 – 17.30.

Oppgavesettet er på 4 sider.

Vedlegg: Ingen.

Tillatte hjelpemidler: Alle trykne og skrevne.

Kontroller at oppgavesettet er komplett
før du begynner å besvare spørsmålene.

Oppgave 1.

Betrakt følgende tilstandsmodell (“state space model”):

$$\begin{aligned}x_t &= \phi x_{t-1} + \varepsilon_t && \text{tilstandsligning} \\y_t &\sim \text{Poisson}(\exp\{1 + x_t\}) && \text{observasjonsligning}\end{aligned}$$

der x_0 og $\varepsilon_1, \varepsilon_2, \dots$ er uavhengige og standard normal fordelte. Vi ønsker å estimere ϕ basert på observasjoner y_1, \dots, y_T . Vi vil gjøre dette Bayesiansk og anta vi har en a priori fordeling $N(0, \sigma_\psi^2)$ på ϕ .

En mulig måte å estimere ϕ i slike situasjoner er å utvide tilstandsmodellen til følgende modell:

$$\begin{aligned}\phi_t &= \phi_{t-1} && \text{tilstandsligning 1} \\x_t &= \phi_{t-1} x_{t-1} + \varepsilon_t && \text{tilstandsligning 2} \\y_t &\sim \text{Poisson}(\exp\{1 + x_t\}) && \text{observasjonsligning}\end{aligned}$$

der $\phi_0 \sim N(0, \sigma_\psi^2)$. Ikke-lineære filtre prøver å beregne a posteriori fordelingen til (ϕ_t, x_t) basert på y_1, \dots, y_t . Siden $\phi = \phi_T$ vil dermed a posteriori fordelingen til (ϕ_T, x_T) basert på y_1, \dots, y_T gi oss a posteriori fordelingen til ϕ gitt y_1, \dots, y_T .

(Fortsettes side 2.)

Simuleringsmetoder for ikke-lineære filtre kan dermed brukes på den *bivariate* tilstandsvektoren (ϕ_t, x_t) .

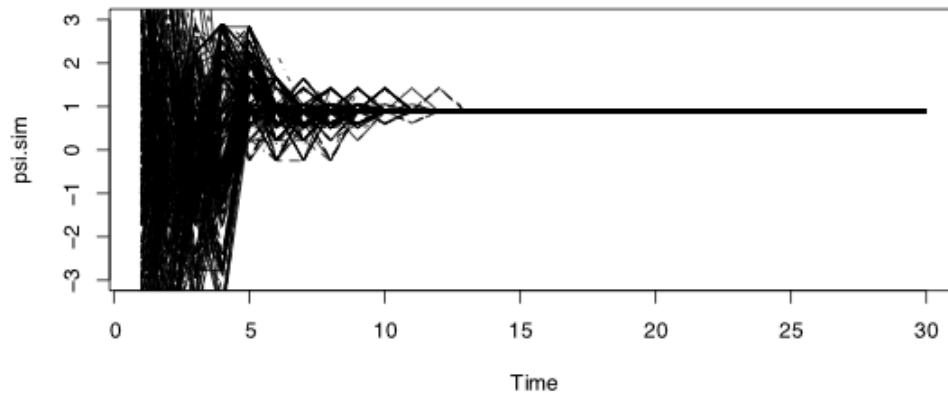
- (a) Forklar de generelle prinsipper bak “sequential importance sampling” (SIS).

Diskuter hvorfor resampling generelt er viktig i forbindelse med SIS algoritmer.

- (b) Simuleringer fra a posteriori fordelingen til (ϕ_t, x_t) basert på y_1, \dots, y_t ble utført ved følgende SIS algoritme:

- Trekk \tilde{x}_t^j fra $N(\phi_{t-1}^j x_{t-1}^j, 1)$ for $j = 1, \dots, M$
- Sett $\tilde{\psi}_t^j = \psi_{t-1}^j$ for $j = 1, \dots, M$.
- Beregn vektene $w_t^j = p(y_t | x_t = \tilde{x}_t^j)$ for $j = 1, \dots, M$ og de normaliserte vektene $q_t^j = w_t^j / \sum_{j'} w_t^{j'}$.
- Trekk $(x_t^1, \psi_t^1), \dots, (x_t^M, \psi_t^M)$ fra $\{(\tilde{x}_t^1, \tilde{\psi}_t^1), \dots, (\tilde{x}_t^M, \tilde{\psi}_t^M)\}$ med tilbakelegging og med sannsynligheter q_t^1, \dots, q_t^M .

Figuren nedenfor viser simuleringer av ψ_t for $t = 1, \dots, T$ basert på en SIS algoritme *med* resampling. Hver kurve svarer til en sekvensen av simulerte ϕ 'er, $\phi_1^j, \dots, \phi_T^j$. De ulike simuleringer $\phi_t^j, j = 1, \dots, M$ for en fast t er (tilnærmet) fra a posteriori fordelingen til ϕ_t gitt y_1, \dots, y_t . Her er $T = 30$ og $M = 50$.



Hvorfor blir det færre og færre forskjellige verdier av simulerte ψ 'er etterhvert? Hvilke problemer skaper dette for estimering av ψ ?

- (c) En mer effektiv algoritme kan oppnås ved å integrere ut den ukjente ψ under simulering av x -prosessen.

En kan vise (du behøver ikke å vise dette) at

$$p(\psi | x_1, \dots, x_t, y_1, \dots, y_t) = N(\hat{\psi}_t, \sigma_t^2)$$

(Fortsettes side 3.)

der

$$\hat{\psi}_t = \frac{\sigma_\psi^2 \sum_{i=2}^t x_i x_{i-1}}{1 + \sigma_\psi^2 \sum_{i=2}^t x_{i-1}^2}$$

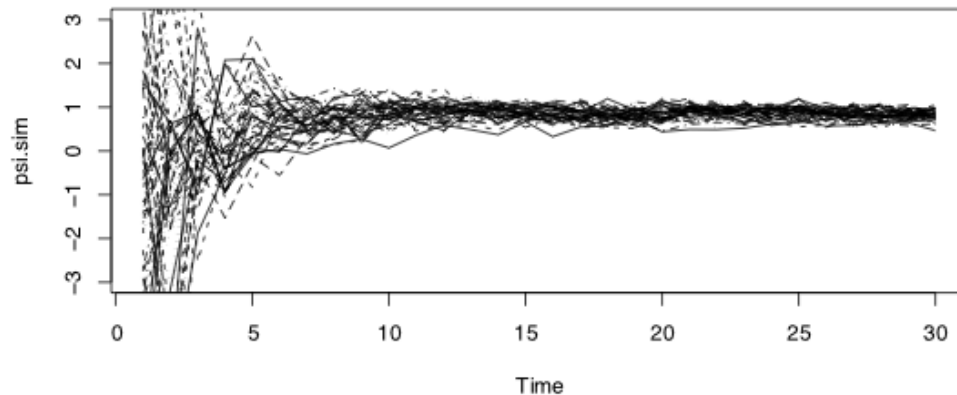
$$\sigma_t^2 = \frac{\sigma_\psi^2}{1 + \sigma_\psi^2 \sum_{i=2}^t x_{i-1}^2}$$

Bruk dette til å finne fordelingen $p(x_{t+1}|x_1, \dots, x_t, y_1, \dots, y_t)$.

(d) Betrakt nå SIS algoritmen (med resampling) som på tidspunkt t går gjennom følgende trinn:

- Trekk \tilde{x}_t^j fra $p(x_t|x_1, \dots, x_{t-1}, y_1, \dots, y_{t-1})$ for $j = 1, \dots, M$.
- Beregn vektene $w_t^j = p(y_t|x_t = \tilde{x}_t^j)$ for $j = 1, \dots, M$ og de normaliserte vektene $q_t^j = w_t^j / \sum_{j'} w_t^{j'}$.
- Trekk x_t^1, \dots, x_t^M fra $\{\tilde{x}^1, \dots, \tilde{x}^M\}$ med tilbakelegging og med sannsynligheter q_t^1, \dots, q_t^M .
- Trekk $\psi_t^j \sim p(\psi|x_1^j, \dots, x_t^j, y_1, \dots, y_t)$

Figuren nedenfor viser simuleringer av ψ_t basert på denne algoritmen.



Hvilke fordeler har denne algoritmen i forhold til den gitt i (b)?

For å kunne estimere a posteriori forventningen til ψ , er det da nødvendig å simulere ψ -ene i det hele tatt? Hvis ikke, vis hvordan en slik estimering kan skje. Hva kalles denne teknikken?

Oppgave 2.

Betrakt følgende algoritme, som vi vil kalle Barker's algoritme (etter Barker (1965) som foreslo denne):

Gitt nåværende tilstand $\mathbf{x}^{(t)}$:

- Trekk \mathbf{y} fra forslagsfordelingen $T(\mathbf{x}^{(t)}, \mathbf{y})$.
- Trekk $U \sim \text{Uniform}[0, 1]$ og oppdater

$$\mathbf{x}^{(t+1)} = \begin{cases} \mathbf{y}, & \text{hvis } U \leq r_B(\mathbf{x}^{(t)}, \mathbf{y}) \\ \mathbf{x}^{(t)} & \text{ellers} \end{cases}$$

der

$$r_B(\mathbf{x}, \mathbf{y}) = \frac{\pi(\mathbf{y})T(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})T(\mathbf{x}, \mathbf{y}) + \pi(\mathbf{y})T(\mathbf{y}, \mathbf{x})}.$$

Vi vil anta at $T(\mathbf{x}, \mathbf{y}) > 0$ for alle \mathbf{x}, \mathbf{y} .

- Vis at $\{\mathbf{x}_t\}$ er en Markov kjede som konvergerer (i fordeling) mot $\pi(\mathbf{x})$.
- Forklar hvordan vi kan bruke simuleringene $\{\mathbf{x}^{(t)}\}$ til å estimere $E_\pi f = \int_{\mathbf{x}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$.

Hva slags egenskaper ved Markov kjeden påvirker presisjonen av et slikt estimat?

Anta A_1 og A_2 er to transisjonskjerner for Markov kjeder med den samme stasjonærfordeling π . La v_1 være variansen til estimatet på $E_\pi f$ basert på simuleringer ved hjelp av A_1 og v_2 være variansen til estimatet på $E_\pi f$ basert på simuleringer ved hjelp av A_2 .

Anta $A_1(\mathbf{x}, \mathbf{y}) \geq A_2(\mathbf{x}, \mathbf{y})$ for alle $\mathbf{y} \neq \mathbf{x}$. En kan da vise at $v_1 \leq v_2$ (dette behøver du ikke å vise).

- La

$$r_M(\mathbf{x}, \mathbf{y}) = \min \left\{ 1, \frac{\pi(\mathbf{y})T(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})T(\mathbf{x}, \mathbf{y})} \right\}.$$

Vis at $r_M(\mathbf{x}, \mathbf{y}) \geq r_B(\mathbf{x}, \mathbf{y})$ for alle \mathbf{x}, \mathbf{y} .

Bruk dette til å argumentere for at Metropolis-Hastings algoritmen er mer effektiv enn Barkers algoritme.

Basert på forskjellene på de to algoritmene, synes du dette er et rimlig resultat?

SLUTT