Royal Institute of Technology

STAT. METH. IN CS –EXACT INFERENCE, SAMPLING, & PARTICLE FILTERS

Lecture 9

DICE CONSTRUCTION

CHAP. 23 MONTE CARLO INFERENCE - SAMPLING

X outcome of dice

$$\mu = \sum_x x p(x)$$

$$X_1, \dots, X_N \sim p(X)$$

⇓

$$E\left[\frac{1}{N}\sum_{n=1}^{N} X_n\right] = \frac{1}{N}\sum_{n=1}^{N} E\left[X_n\right] = \mu$$

Can also be used for posterior

$$X_1, \dots, X_N \sim p(X|D)$$

MONTE CARLO BASICS

By sampling (↓ sample size)

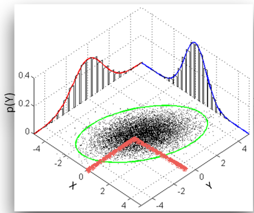
$$Z_{1:n}^s \sim \pi_n(Z_{1:n}^s),\ s \in [S]$$

we get the approximation

$$\pi_n(Z_{1:n}) \approx \tilde{\pi}_n(Z_{1:n}^s) := \frac{1}{S}\sum_{s\in[S]} \delta_{Z_{1:n}^i}(Z_{1:n})$$

where δ is the Dirac delta mass, i.e.,

$$\delta_{X_0}(X) = \begin{cases} 1 & \text{if } X = X_0 \\ 0 & \text{otherwise} \end{cases}$$



MONTE CARLO BASICS - ESTIMATING EXPECTATION

If φ_n is a real valued function of the r.v. $Z_{1:n}$ and

$$Z^s_{1:n} \sim \pi_n(Z^s_{1:n}),\ s \in [S]$$

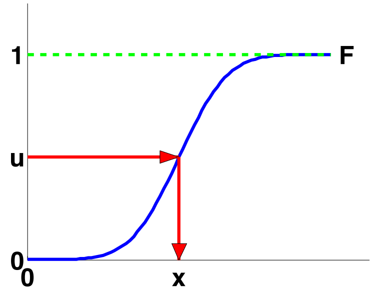
then

$$E_{\pi_n}[\varphi_n(Z_{1:n})] = \int \varphi_n(z_{1:n})\pi_n(z_{1:n})\, dz_{1:n}$$

is approximated by

$$I_n^{\mathrm{MC}}(\varphi_n) = \int \varphi_n(z_{1:n})\tilde{\pi}_n(z_{1:n})\, dz_{1:n}$$
$$= \frac{1}{S}\sum_{s\in[S]} \varphi(Z^s_{1:n})$$

HOW TO SAMPLE FROM A DISTRIBUTION?



If we can compute F, F^{-1} can be found with binary search

Theorem

For cdf F, if u~U(0,1), then $F^{-1}(u)$~F

Proof: $p(F^{-1}(u) \leq x) = p(u \leq F(x)) = F(x)$

So, to sample from F

sample u~U(0,1)

output $F^{-1}(u)$

EX.

Pdf

$$p(X|\lambda) = \lambda e^{-\lambda x}$$

Cdf

$$F(X|\lambda) = 1 - e^{-\lambda x}$$

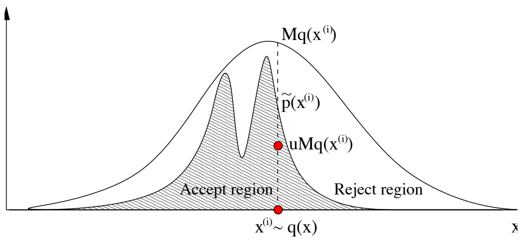
Inverse

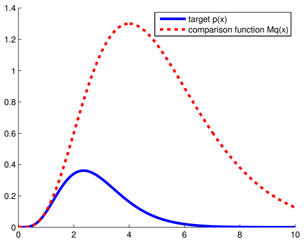
$$F^{-1}(p) = -\frac{\ln(1-p)}{\lambda}$$

REJECTION SAMPLING

- We want to sample from Beta(2,2)
  - pdf f(x) is proportional to x(1-x)
  - notice f has max at 1/2 and where it is 1/4

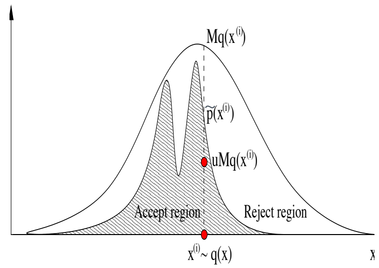
REJECTION SAMPLING





- Want to sample from p, can compute $\tilde{p}$ where $p(x)=p(\tilde{x})/Z_p$
- Use proposal distribution q(x) s/t $Mq(x) \geq \tilde{p}(x)$ (for known M)

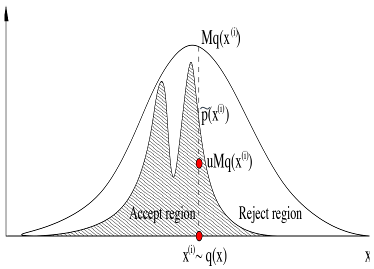
REJECTION SAMPLING



Algorithm

- sample x~q(x)
- sample u~U(0,1)
- if $uMq(x) \leq \tilde{p}(x)$, accept (and output x)

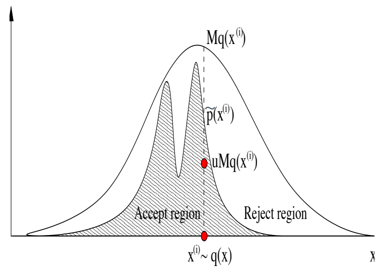
CORRECT CDF



Let $$S = \{(x,u) : u \leq \tilde{p}(x)/(Mq(x))\} = \{(x,u) : uMq(x) \leq \tilde{p}(x)\}$$

$$p(x \leq x_0, x \text{ accepted}) = \int_{-\infty}^{x_0} p_{u\sim U(0,1)}\left((x,u) \in S\right) q(x)\mathrm{d}x$$
$$= \int_{-\infty}^{x_0} p_{u\sim U(0,1)}\left(u \leq \tilde{p}(x)/Mq(x)\right) q(x)\mathrm{d}x$$
$$= \int_{-\infty}^{x_0} \frac{\tilde{p}(x)}{Mq(x)} q(x)\mathrm{d}x = \frac{1}{M}\int_{-\infty}^{x_0} \tilde{p}(x)\mathrm{d}x$$

CORRECT CDF



Let $$S = \{(x,u) : u \leq \tilde{p}(x)/(Mq(x))\} = \{(x,u) : uMq(x) \leq \tilde{p}(x)\}$$

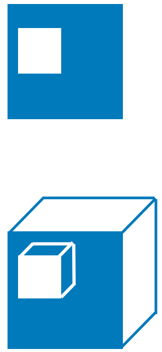
$$p(x \leq x_0 | x \text{ accepted}) = \frac{p(x \leq x_0, x \text{ accepted})}{p(x \text{ accepted})} = \frac{\frac{1}{M}\int_{-\infty}^{x_0} \tilde{p}(x)\mathrm{d}x}{\frac{1}{M}\int_{-\infty}^{\infty} \tilde{p}(x)\mathrm{d}x}$$

$$= \frac{Z_p \int_{-\infty}^{x_0} p(x)\mathrm{d}x}{Z_p \int_{-\infty}^{\infty} p(x)\mathrm{d}x} = \int_{-\infty}^{x_0} p(x)\mathrm{d}x$$

REJECTION SAMPLING

- We want to sample from Beta(2,2)
  - pdf is proportional to f(x)=x(1-x)
  - notice f has max at 1/2 and where it is 1/4
- To sample from Beta(2,2)
  - Sample x~ Uni(0,1)
  - Sample u ~ Uni(0,1)
  - If x(1-x) < u/4 accept x (output it) otherwise try again

HIGHER DIMENSIONS



$c<1$, $c \rightarrow c^D$

- Rejection sampling performs poorly in higher dimensions

IMPORTANCE SAMPLING

Want to compute

$$E_{p(x)}\left[f(x)\right] = \int f(x)p(x)\mathrm{d}x$$

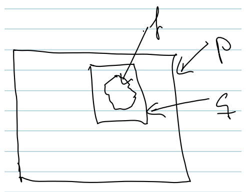
Sample $x^1,\ldots,x^S \sim q(x)$ Let $w_s=p(x^s)/q(x^s)$

Compute $\hat{I} = \frac{1}{S}\sum_{s=1}^{S} w_s f(x^s)$

Notice

$$E\left[\hat{I}\right] = \int f(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x = E_{p(x)}\left[f(x)\right]$$

- We can sample from q, but not from p
- Or q is better



IMPORTANCE SAMPLING - UNNORMALIZED DISTRIBUTIONS

Want to compute

$$E_{p(x)}\left[f(x)\right] = \int f(x)p(x)\mathrm{d}x$$

Sample $x^1,\ldots,x^S \sim q(x)$ Let $\tilde{w}_s=\tilde{p}(x^s)/\tilde{q}(x^s)$

Notice

$$E_{p(x)}\left[f(x)\right] = \int f(x)p(x)\mathrm{d}x = \frac{Z_q}{Z_p}\int f(x)\frac{q(x)}{\tilde{q}(x)}\tilde{p}(x)\mathrm{d}x \approx \frac{Z_q}{Z_p}\frac{1}{S}\sum_{s=1}^{S}\tilde{w}_s f(x^s)$$

Pointwise

- We can compute and sample from $\tilde{q}$, and compute $\tilde{p}$
- $p(x)=\tilde{p}(x)/Z_p$
- $q(x)=\tilde{q}(x)/Z_q$

Let $\tilde{w}_s=\tilde{p}(x^s)/\tilde{q}(x^s)$

$$E_{p(x)}\left[f(x)\right] \approx \frac{Z_p}{Z_q}\frac{1}{S}\sum_{s=1}^{S}\tilde{w}_s f(x^s)$$

Now notice

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(x)\mathrm{d}x \qquad \text{Integration}$$

$$= \int \tilde{p}(x)\frac{q(x)}{\tilde{q}(x)}\mathrm{d}x \qquad \text{Pointwise}$$

$$\approx \frac{1}{S}\sum_{s=1}^{S}\tilde{w}_s$$

IMPORTANCE SAMPLING - UNNORMALIZED DISTRIBUTIONS

- We can compute and sample from $\tilde{q}$, and compute $\tilde{p}$
- $p(x)=\tilde{p}(x)/Z_p$
- $q(x)=\tilde{q}(x)/Z_q$

Let $w_s=\tilde{p}(x^s)/\tilde{q}(x^s)$

$$E_{p(x)}\left[f(x)\right] \approx \frac{Z_p}{Z_q}\frac{1}{S}\sum_{s=1}^{S}\tilde{w}_s f(x^s)$$

and

$$\frac{Z_q}{Z_p} \approx \frac{1}{S}\sum_{s=1}^{S}\tilde{w}_s$$

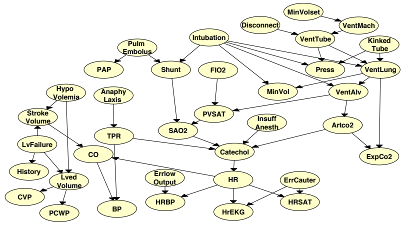
So

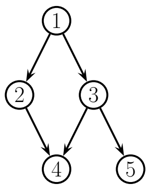
$$E_{p(x)}\left[f(x)\right] \approx \frac{\sum_{s=1}^{S}\tilde{w}_s f(x^s)}{\sum_{s=1}^{S}\tilde{w}_s}$$

IMPORTANCE SAMPLING - UNNORMALIZED DISTRIBUTIONS

- We can compute and sample from $\tilde{q}$, and compute $\tilde{p}$
- $p(x)=\tilde{p}(x)Z_p$
- $q(x)=\tilde{q}(x)Z_q$
- Biased!

IMPORTANCE SAMPLING FOR DGM





- Easy to sample from $p(x|G)$ - sample roots and continue in topological order
- How to sample from $p(x_H|x_O=y,G)$?
  - "logic sampling" as above, reject if $x_o \neq y$
  - not efficient, impossible for continuous values

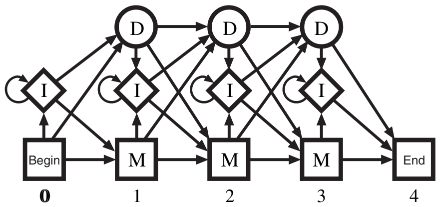
LIKELIHOOD WEIGHTING (IMPORTANCE)

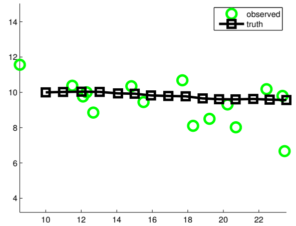
- Sampling
  - sample X_H
  - do not sample X_O, use y
- So, proposal

$$q(\boldsymbol{X}) = \prod_{u\in H} p(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})\prod_{u\in O} I(X_u=y_u)$$

- & weights (for X from q)

$$w(X) = \frac{p(X)}{q(X)} = \frac{\prod_{u\in H} p(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})\prod_{u\in O} p(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})}{\prod_{u\in H} p(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})\prod_{u\in O} 1} = \prod_{u\in O} p(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})$$





CONTINUOUS STATES

STATE SPACE MODEL SSM

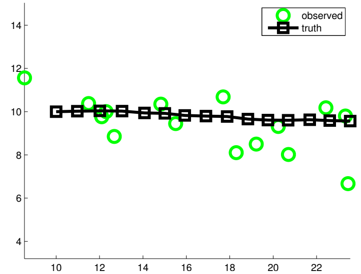
Generic form

transition

$$z_t := g(u_t, z_{t-1}, \epsilon_t)$$

observation

$$y_t := h(z_t, u_t, \delta_t)$$



STATE SPACE MODEL SSM

Generic form

hidden state → input → system noise

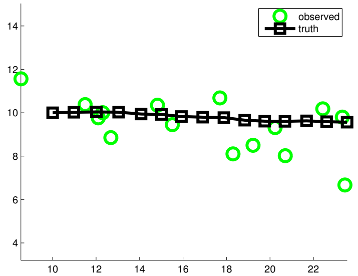
$$z_t := g(u_t, z_{t-1}, \epsilon_t)$$

observation

$$y_t := h(z_t, u_t, \delta_t)$$

observation noise

Estimate belief state (filtering) $p(z_t | y_{1:t}, u_{1:t}, \theta)$



STATE SPACE MODEL SSM

Generic form

hidden state → input → system noise

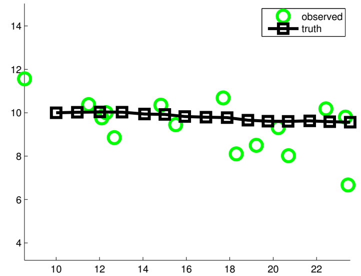
$$z_t := g(u_t, z_{t-1}, \epsilon_t)$$

observation

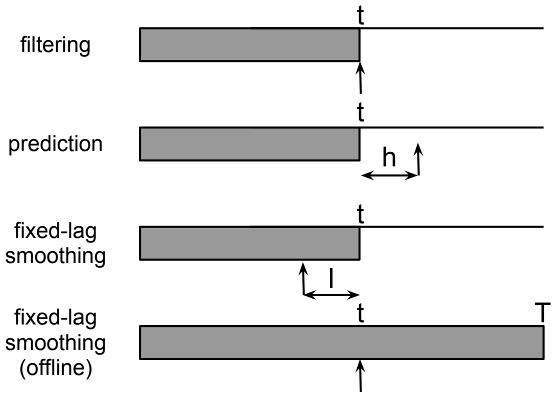
$$y_t := h(z_t, u_t, \delta_t)$$

observation noise

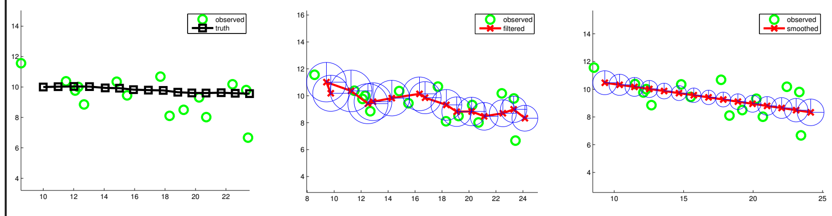
Estimate belief state (filtering) $p(z_t | y_{1:t}, u_{1:t}, \theta)$



COMPUTATIONAL PROBLEMS



FILTERING & SMOOTHING



Filtering

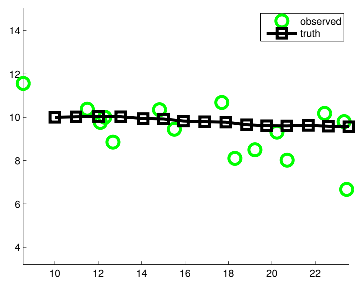
Smoothing

LG-SSM EXAMPLE

An object moves in a 2D plane

★ coordinates z_{1t}, z_{2t}

★ velocity $\dot{z}_{1t}, \dot{z}_{2t}$



LG-SSM EXAMPLE

An object moves in a 2D plane

★ coordinates (z_{1t}, z_{2t})

★ velocity $(\dot{z}_{1t}, \dot{z}_{2t})$

★ sampling period Δ

★ system noise

$\epsilon_t \sim \mathcal{N}(0, Q)$

★ observation noise

$\delta_t \sim \mathcal{N}(0, Q)$

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t$$

$$\begin{pmatrix} z_{1t} \\ z_{2t} \\ \dot{z}_{1t} \\ \dot{z}_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_{1,t-1} \\ z_{2,t-1} \\ \dot{z}_{1,t-1} \\ \dot{z}_{2,t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix}$$

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{z}_t + \boldsymbol{\delta}_t$$

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} z_{1t} \\ z_{2t} \\ \dot{z}_{1t} \\ \dot{z}_{2t} \end{pmatrix} + \begin{pmatrix} \delta_{1t} \\ \delta_{2t} \\ \delta_{3t} \\ \delta_{4t} \end{pmatrix}$$

LG-SSM EXAMPLE

An object moves in a 2D plane

★ coordinates (z_{1t}, z_{2t})

★ velocity $(\dot{z}_{1t}, \dot{z}_{2t})$

★ sampling period Δ

★system noise

$\epsilon_t \sim \mathcal{N}(0, Q)$

★observation noise

$\delta_t \sim \mathcal{N}(0, Q)$

Initial state

$$p(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1 | \boldsymbol{\mu}_{1|0}, \Sigma_{1|0})$$

Initial covariance matrix "broad"

Kalman filtering gives belief states

$$p(\boldsymbol{z}_t | \boldsymbol{y}_{1:t})$$

Kalman smoothing gives belief states

$$p(\boldsymbol{z}_t | \boldsymbol{y}_{1:T})$$

CONSIDER SIMPLE LG-SSM

Transition

$$z_t = a z_{t-1} + \epsilon_t$$

z_t scalar & $\epsilon_t \sim \mathcal{N}(0, Q)$

Observation

$$y_t = c z_t + \delta_t$$

where

$\delta_t \sim \mathcal{N}(0, R)$

Prediction step:

$$p(z_t | y_{1:t-1}) = \int_{z_{t-1}} \mathcal{N}(z_t | a z_{t-1}, Q) \mathcal{N}(z_{t-1} | \mu_{t-1}, \sigma^2_{t-1}) \, dz_{t-1}$$

where

μ_{t-1} is the filtered expectation on z_{t-1}

σ^2_{t-1} is the filtered variance on z_{t-1}

this is Gaussian (similar to Gaussian being self conjugate)

Measurement step (also gives Gaussian):

$$p(z_t | y_{1:t}) \propto p(y_t | z_t) p(z_t | y_{1:t-1})$$

COMPLEXITY, SMOOTHING, & PARAMETERS

- Kalman filtering for MVN involves matrix inversion
- Complexity $O(|y|^3 + |z|^2)$
- Kalman smoothing gives

  $p(\boldsymbol{z}_t | \boldsymbol{y}_{1:T})$
- EM can be used to find parameters

THE END