Probabilistic Graphical Models

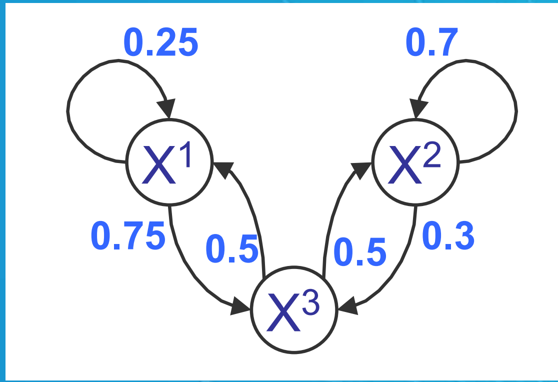
Advanced MCMC Methods:

Optimization + MCMC

Eric Xing

Lecture 10, February 12, 2020

Reading: https://arxiv.org/pdf/1206.1901.pdf



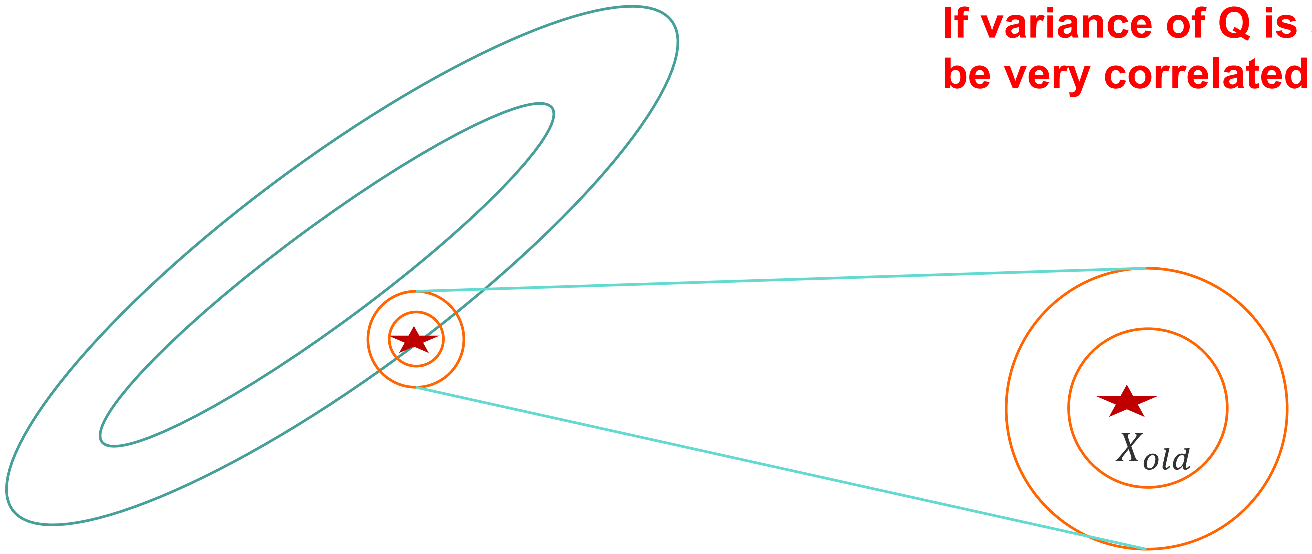
Random walk in MCMC

Might reject a lot of samples

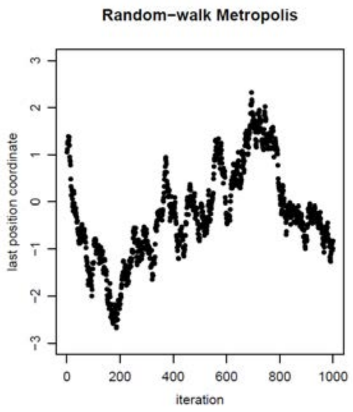
$P(X)$ $Q(X_{new}|X_{old})$

$$\min\{\ 1,\quad \frac{P(X_{new})Q(X_{old}|X_{new})}{P(X_{old})Q(X_{new}|X_{old})}\}$$

Random walk in MCMC



If variance of Q is small then next sample might be very correlated to the previous one



$P(X)$

$Q(X_{new}|X_{old})$

$$\min\{\ 1,\ \frac{P(X_{new})Q(X_{old}|X_{new})}{P(X_{old})Q(X_{new}|X_{old})}\}$$

MCMC: Recap

- Random walk can have poor acceptance rate
- The samples can have high correlation between themselves reducing the effective sample size
- Can we have a better proposal
  - Using gradient information
  - Using approximation of the given probability distribution

Hamiltonian Monte Carlo

- Hamiltonian Dynamics (1959)
  - Deterministic System
- Hybrid Monte Carlo (1987)
  - United MCMC and molecular Dynamics
- Statistical Application (1993)
  - Inference in Neural Networks
  - Improves acceptance rate
  - Uncorrelated Samples

Target distribution:

$$P(x) = \frac{e^{-E(x)}}{Z}$$

The Hamiltonian:

$$H(x,p) = E(x) + K(p)$$

$$\dot{x} = p \quad \dot{p} = -\frac{\partial E(x)}{\partial x} \qquad K(p) = p^{\mathrm{T}}p/2$$

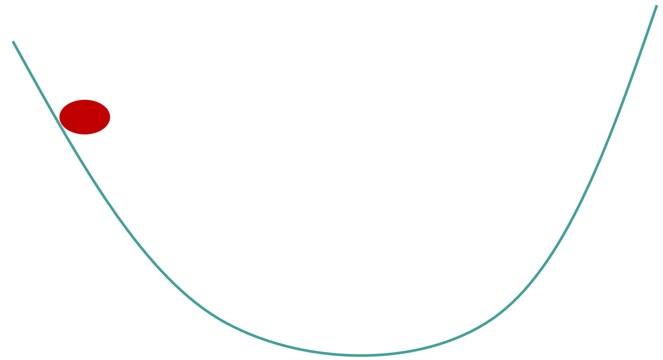
Auxiliary distribution:

$$P_H(x,p) = \frac{e^{-E(x)-K(p)}}{Z_H}$$

Hamiltonian Dynamics

- Position vector x, Momentum vector p
- Kinetic Energy $K(p)$
- Potential Energy $U(x)$
- Define $H(p, x) = K(p) + U(x)$

Hamiltonian Dynamics



- Position vector x, Momentum vector p
- Kinetic Energy $K(p)$
- Potential Energy $U(x)$
- Define $H(p,x) = K(p) + U(x)$
- Hamiltonian Dynamics
  - Can help getting gradient of U over x to draw next sample!

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}$$

Alternative notation

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

Hamiltonian Dynamics: Example

- Kinetic Energy $K(p) = \frac{|p|^2}{2}$
- Potential Energy $U(q) = \frac{q^2}{2}$
- So

$$\frac{dq}{dt} = p, \qquad \frac{dp}{dt} = -q$$

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

- And

$$q(t) = r\cos(a+t), \quad p(t) = -r\sin(a+t)$$

How to compute updates: Euler's Method

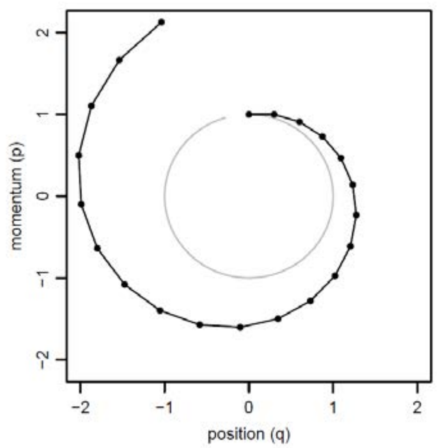
$$p_i(t+\varepsilon) = p_i(t) + \varepsilon \frac{dp_i}{dt}(t) = p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q(t))$$

$$q_i(t+\varepsilon) = q_i(t) + \varepsilon \frac{dq_i}{dt}(t) = q_i(t) + \varepsilon \frac{p_i(t)}{m_i}$$

$$\begin{pmatrix} p(t+\epsilon) \\ q(t+\epsilon) \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & 1 \end{pmatrix} \begin{pmatrix} p(t) \\ q(t) \end{pmatrix}$$

A divergent series!

(a) Euler's Method, stepsize 0.3



How to compute updates: Leapfrog Method

- The updates looks like

$$p_i(t+\varepsilon/2) = p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t))$$

$$q_i(t+\varepsilon) = q_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i}$$

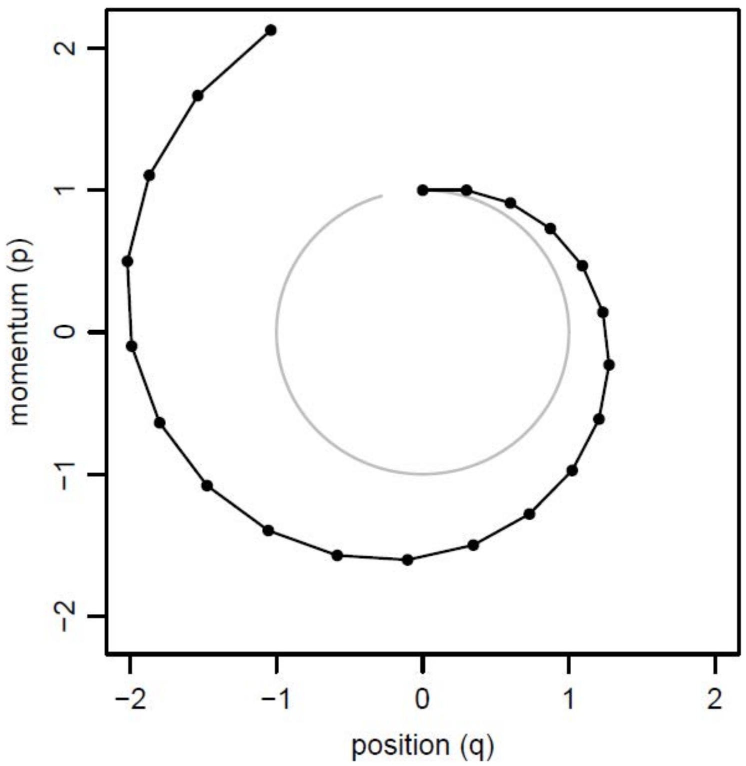
$$p_i(t+\varepsilon) = p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t+\varepsilon))$$

$$\begin{pmatrix} p(t+\epsilon) \\ q(t+\epsilon) \end{pmatrix} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p(t+\epsilon/2) \\ q(t+\epsilon) \end{pmatrix}$$

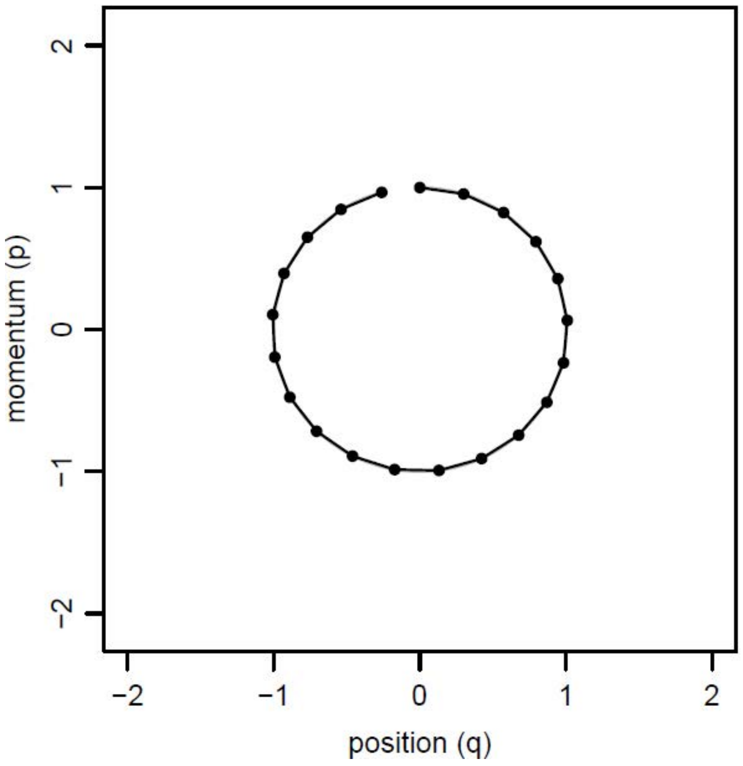
A shear transformation → volume preserving

Leapfrog Vs Euler

(a) Euler's Method, stepsize 0.3



(c) Leapfrog Method, stepsize 0.3



$$q(t) = r\cos(a+t), \quad p(t) = -r\sin(a+t)$$

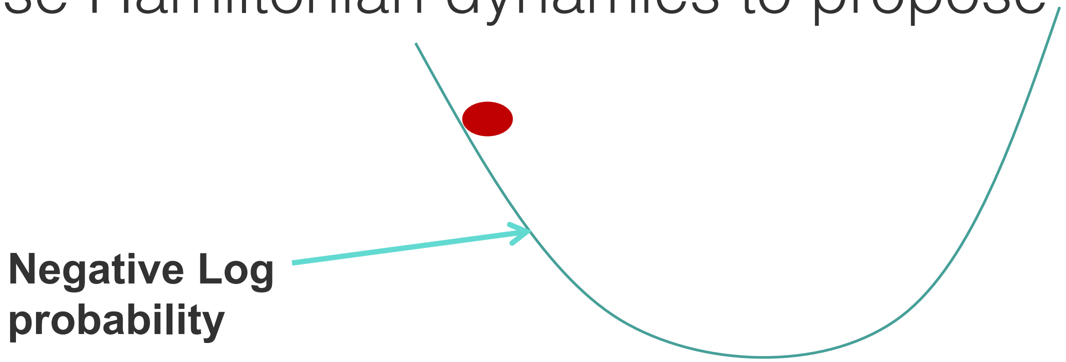
MCMC from Hamiltonian Dynamics

- Let q be variable of interest (e.g., latent parameters of a model)
- Define:

$$P(q,p) = \frac{1}{Z}\exp(-U(q)/T)\exp(-K(p)/T)$$

  - where $U(q) = -\log\left[\pi(q)L(q|D)\right]$ $\quad K(p) = \sum_{i=1}^{d}\frac{p_i^2}{2m_i}$
  - $\pi(q)$ denotes the prior, and $L((q|D)$ denotes the data likelihood

- Key Idea: Use Hamiltonian dynamics to propose next step.



MCMC from Hamiltonian Dynamics

- Given q_0 (starting state)
- Draw $p \sim N(0,1)$
- Use L steps of leapfrog to propose next state
- Accept / reject based on change in Hamiltonian

Each iteration of the HMC algorithm has two steps. The first changes only the momentum; the second may change both position and momentum. Both steps leave the canonical joint distribution of (q, p) invariant, and hence their combination also leaves this distribution invariant.

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
# Alternate full steps for position and momentum
for (i in 1:L)
{
    q = q + epsilon * p
    if (i!=L) p = p - epsilon * grad_U(q)
}
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
# Alternate full steps for position and momentum
for (i in 1:L)
{
     q = q + epsilon * p
     if (i!=L) p = p - epsilon * grad_U(q)
}
p = p - epsilon * grad_U(q) / 2          p = -p
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
# Alternate full steps for position and momentum
for (i in 1:L)
{
      q = q + epsilon * p
      if (i!=L) p = p - epsilon * grad_U(q)
}
p = p - epsilon * grad_U(q) / 2        p = -p
```

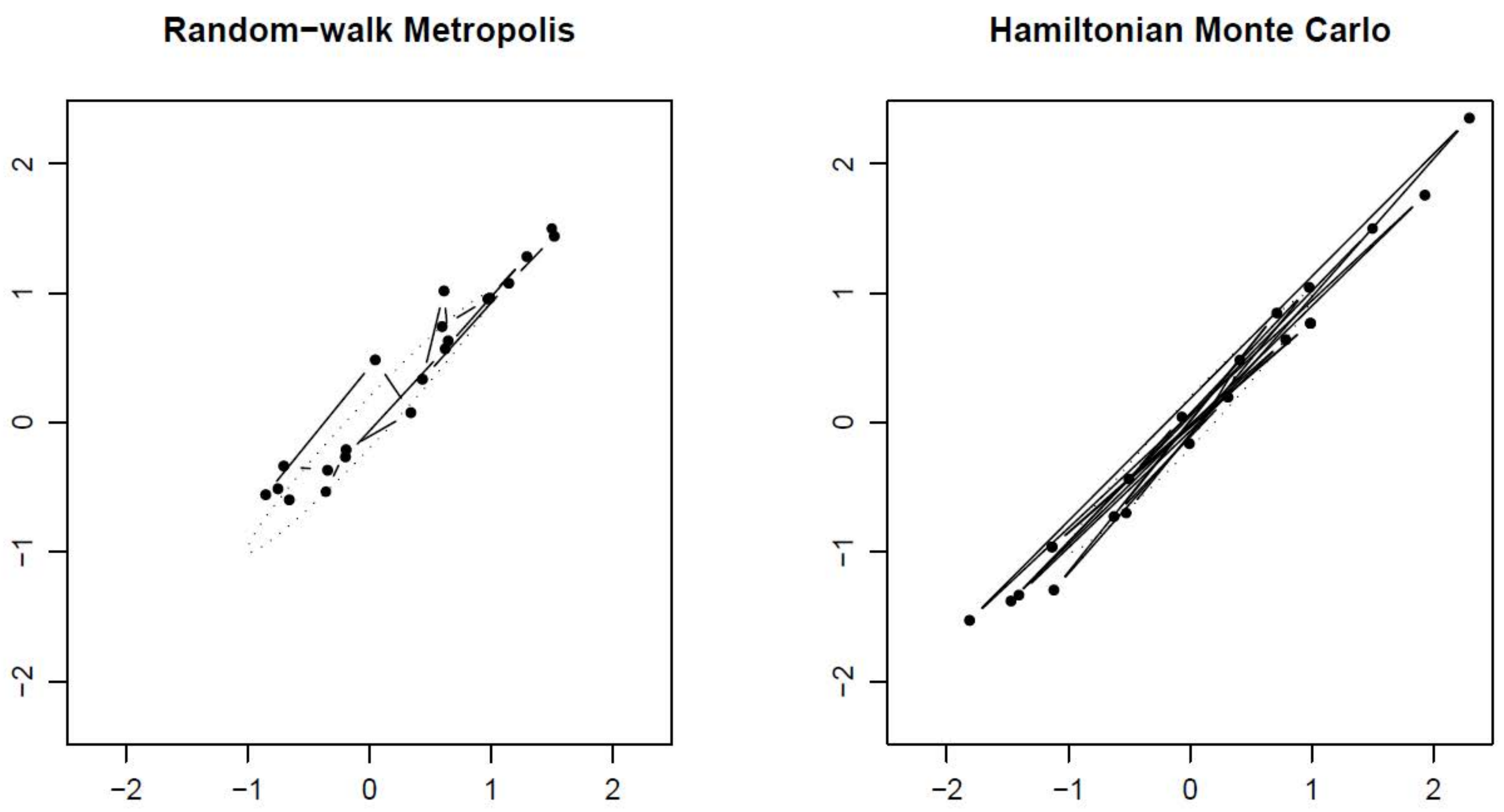
Accept or reject the state at end of trajectory

$$\min\left[1, \exp(-U(q^*) + U(q) - K(p^*) + K(p))\right]$$

MCMC from Hamiltonian Dynamics

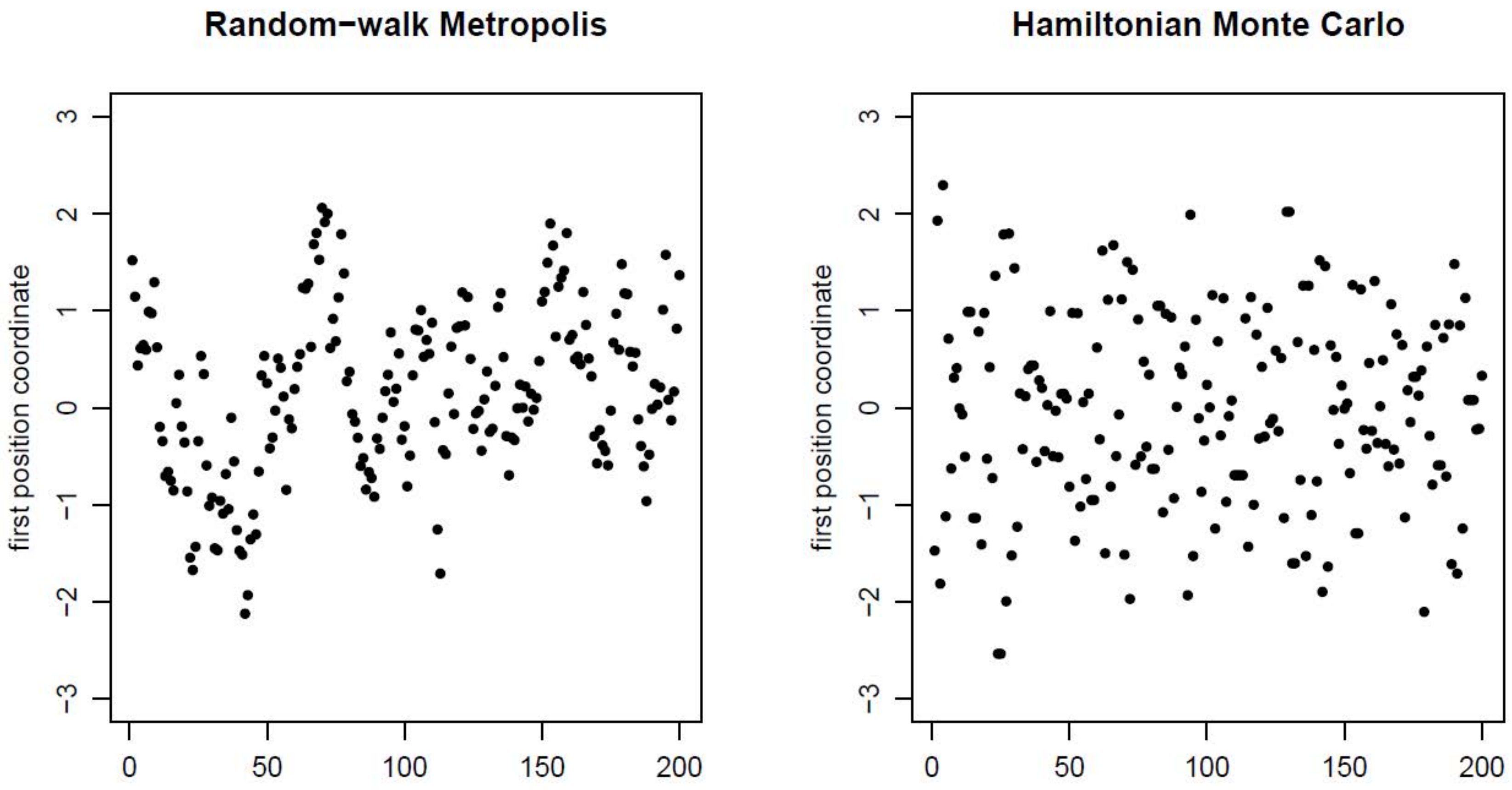
- Detailed balance satisfied
- Ergodic
- canonical distribution invariant

2D Gaussian Example



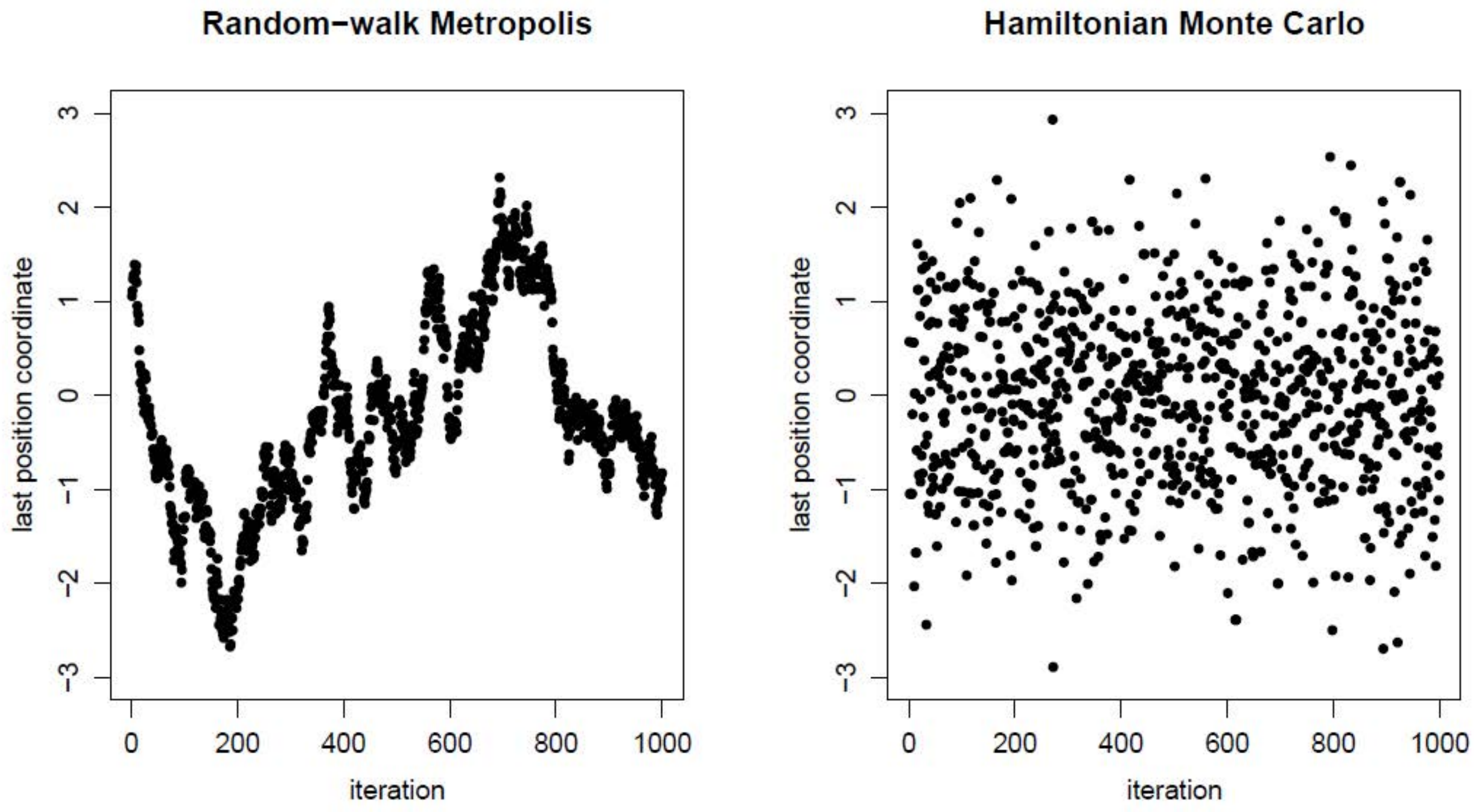
Twenty iterations of the random-walk Metropolis method (with 20 updates per iteration) and of the Hamiltonian Monte Carlo method (with 20 leapfrog steps per trajectory) for a 2D Gaussian distribution with marginal standard deviations of one and correlation 0.98. Only the two position coordinates are plotted, with ellipses drawn one standard deviation away from the mean.

2D Gaussian Example



Two hundred iterations, starting with the twenty iterations shown above, with only the first position coordinate plotted.

100D Gaussian Example



Acceptance Rate

- 2D example HMC : 91% Random Walk: 63%
- 100D example HMC: 87% Random Walk: 25%

Langevin Dynamics

Leapfrog

$$p_i(t+\varepsilon/2) = p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t))$$

$$q_i(t+\varepsilon) = q_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i}$$

$$p_i(t+\varepsilon) = p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t+\varepsilon))$$

One leepfrog step only, all at once:

$$q_i^* = q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i$$

$$p_i^* = p_i - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q^*)$$

accept q^* as the new state with probability

$$\min\left[1,\ \exp\left(-(U(q^*)-U(q)) - \frac{1}{2}\sum_i((p_i^*)^2 - p_i^2)\right)\right]$$

Stochastic Langevin Dynamics

- For large datasets hard to compute the whole gradient

$$q_i^* = q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i$$

$$U(q) = -\log\Big[\pi(q)L(q|D)\Big]$$

Stochastic Gradient Langevin Dynamics

- For large datasets hard to compute the whole gradient

$$q_i^* = q_i - \frac{\varepsilon^2}{2} \underbrace{\frac{\partial U}{\partial q_i}(q)} + \varepsilon p_i$$

Calculate using subset of data

$$U(q) = -\log\Big[\pi(q) L(q|D)\Big]$$

Stochastic Gradient Langevin Dynamics: Bayesian Models

- Posterior $p(\theta|\mathbf{X}) \propto p(\theta) \prod_{i=1}^{N} p(x_i|\theta)$

- SGLD update:

$$\Delta\theta_t = \frac{h_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{ti}|\theta_t)\right) + \eta_t$$

$$\eta_t \sim N(0, h_t)$$

$$q_i^* = q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i$$

$$U(q) = -\log\left[\pi(q) L(q|D)\right]$$

Stochastic Gradient Langevin Dynamics

- High variance in stochastic gradient

- Take help from the optimization community

Conclusion

- HMC can improve acceptance rate and give better mixing
- Stochastic variants can be used to improve performance in large dataset scenarios
- HMC may not be used for discrete variable

Supplementary

Variational MCMC

Sequential Monte Carlo

Towards better proposal

- $Q(X_{new}|X_{old})$ determines when the chain converges
- Idea: Variational approximation of P(X) be the proposal distribution

Variational Inference: Recap

- Interested in posterior of parameters $P(\theta|x)$
- Using Jensen's Inequality

$$\boldsymbol{log(p(x|\theta)} \geq \boldsymbol{E_{q(z)}[log(p(x|\theta)] - E_{q(z)}[log\big(q(z)\big)]}$$

- Choose $q(z|\lambda)$ where λ is the variational parameter
- Replace $p(x|\theta)$ with $p(x|\theta, \xi)$ where ξ is another set of variational parameters
- Using this we can easily obtain un-normalized bound for posterior

$$\boldsymbol{P(\theta|x\,)} \geq \boldsymbol{P^{est}(\theta|x, \lambda, \xi)}$$

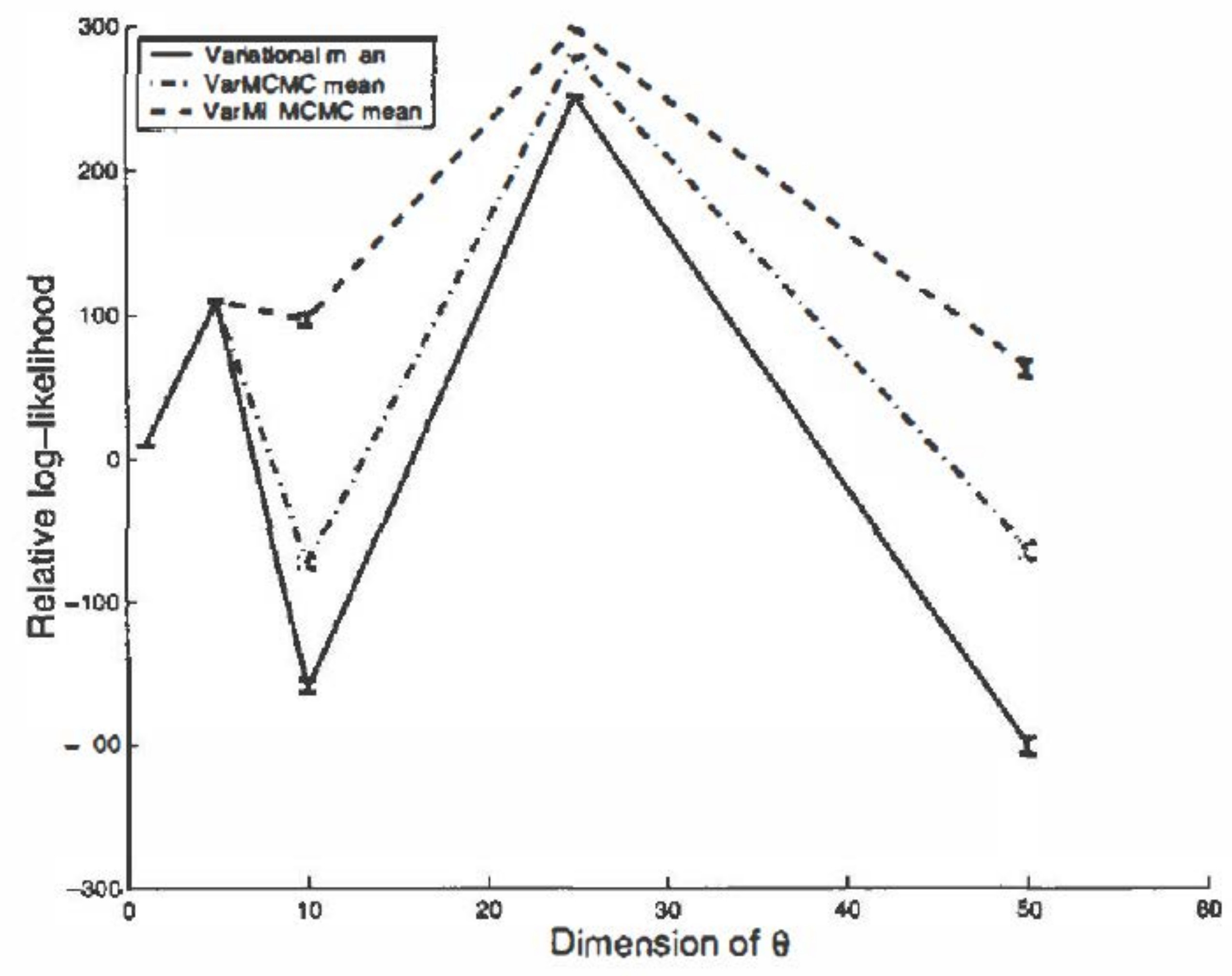
Variational MCMC

- Idea: Variational approximation of P(X) be the proposal distribution
- $Q(\theta_{new}|\theta_{old}) = \boldsymbol{P}^{\boldsymbol{est}}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\xi})$
- Issues:
  - Low acceptance in high dimensions
  - Works well if $\boldsymbol{P}^{\boldsymbol{est}}$ is close to P

Variational MCMC

- Design the proposal in blocks to take care of correlated variables
- Use a mixture of random walk and variational approximation as a proposal distribution
- Now can use stochastic variational methods in estimating $\boldsymbol{P}^{est}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\xi})$

Variational MCMC

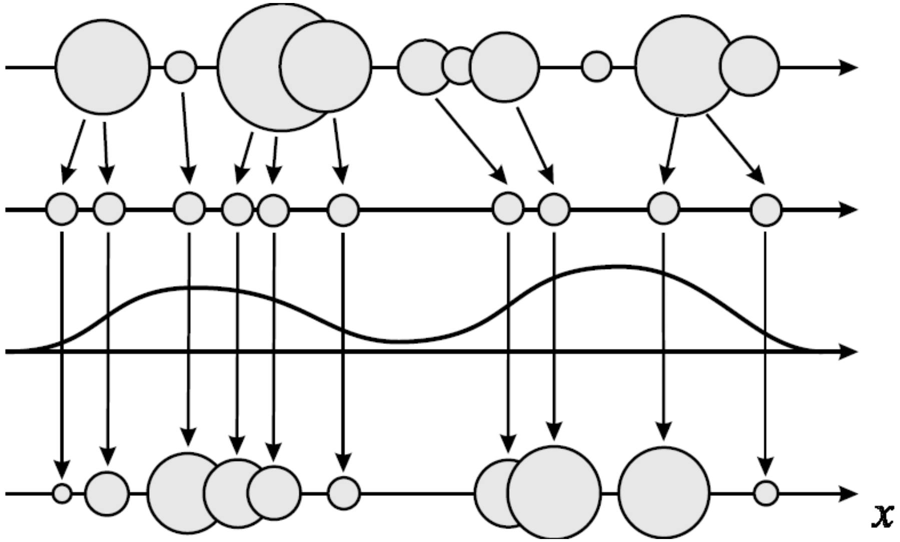


Conclusion

- Adapting proposal distribution can be helpful in
  - Increasing mixing
  - Decreasing time to convergence
  - Increasing acceptance rate
  - Getting uncorrelated information

Recall: weighted resampling

- Sampling importance resampling (SIR):
  1. Draw N samples from Q: $X_1 \dots X_N$
  2. Construct weights: $w_1 \dots w_N$, $\quad w^m = \frac{P(x^m)/Q(x^m)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^m}{\sum_m r^m}$
  3. Sub-sample x from $\{X_1 \dots X_N\}$ w.p. $(w_1 \dots w_N)$



Sequential MC: Sketch of Particle Filters

- The starting point $p(X_t|\mathbf{Y}_{1:t}) = p(X_t \mid Y_t, \mathbf{Y}_{1:t-1}) = \dfrac{p(X_t \mid \mathbf{Y}_{1:t-1})\,p(Y_t \mid X_t)}{\int p(X_t \mid \mathbf{Y}_{1:t-1})\,p(Y_t \mid X_t)\,dX_t}$
  - Thus p(X_t| $Y_{1:t}$) is represented by $\left\{ X_t^m \sim p(X_t \mid \mathbf{Y}_{1:t-1}),\;\; w_t^m = \frac{p(Y_t|X_t^m)}{\sum_{m=1}^{M} p(Y_t|X_t^m)} \right\}$

- A sequential weighted resampler
  - Time update

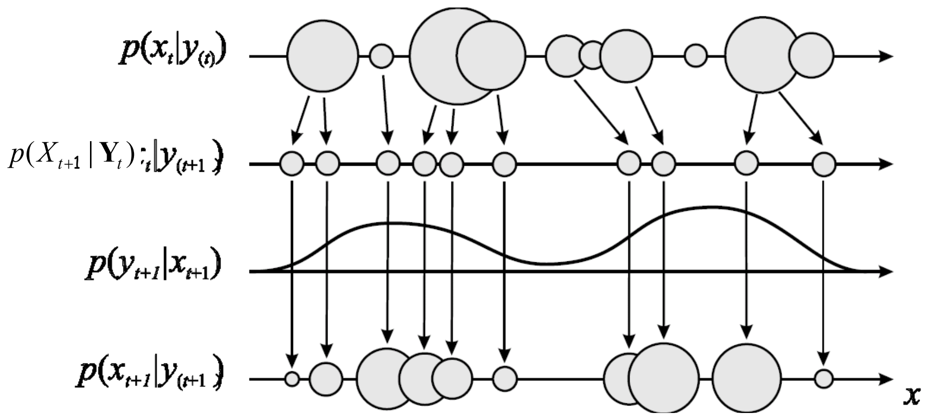
$$p(X_{t+1} \mid \mathbf{Y}_{1:t}) = \int p(X_{t+1} \mid X_t)\,p(X_t \mid \mathbf{Y}_{1:t})\,dX_t$$

$$= \sum_m w_t^m\, p(X_{t+1} \mid X_t^{(m)})$$ (sample from a mixture model)

  - Measurement update

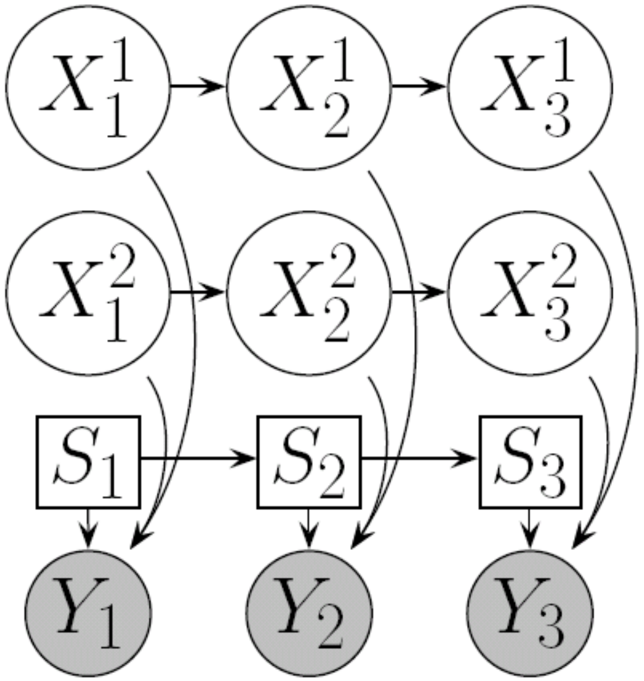
$$p(X_{t+1}|\mathbf{Y}_{1:t+1}) = \frac{p(X_{t+1} \mid \mathbf{Y}_{1:t})\,p(Y_{t+1} \mid X_{t+1})}{\int p(X_{t+1} \mid \mathbf{Y}_{1:t})\,p(Y_{t+1} \mid X_{t+1})\,dX_{t+1}}$$

$$\Rightarrow \left\{ X_{t+1}^m \sim p(X_{t+1} \mid \mathbf{Y}_{1:t}),\;\; w_{t+1}^m = \frac{p(Y_{t+1}|X_{t+1}^m)}{\sum_{m=1}^{M} p(Y_{t+1}|X_{t+1}^m)} \right\}$$ (reweight)



PF for switching SSM

- Recall that the belief state has $O(2^t)$ Gaussian modes



PF for switching SSM

- Key idea: if you knew the discrete states, you can apply the right Kalman filter at each time step.

- So for each old particle *m*, sample $S_t^m \sim P(S_t \mid S_{t-1}^m)$ from the prior, apply the KF (using parameters for S_t^m) to the old belief state $(\hat{x}_{t-1|t-1}^m, P_{t-1|t-1}^m)$ to get an approximation to $P(X_t \mid y_{1:t}, s_{1:t}^m)$

- Useful for online tracking, fault diagnosis, etc.