Learning Objectives

• Become familiar with additional Monte Carlo methods
  – Metropolis-Hastings sampler
  – Component-wise Metropolis-Hastings sampler
    » Generalizes Gibbs sampler to cases in which conditional distributions required for Gibbs sampling are not available
    » Uses rejection sampling to adjust expectations
  – Importance sampling
    » Generalizes Direct Monte Carlo to allow sampling from a distribution other than the target distribution other than the target distribution
    » Uses weights to adjust expectations
  – Sequential importance sampling
    » a.k.a sequential Monte Carlo / particle filter
    » Alternative to Kalman filter to temporal problems when Gaussian linear model is not appropriate
Basic Metropolis Sampler

- Goal: simulate from posterior distribution \( g(\theta | X=x) \)
- Assume: We can calculate prior density \( g(\theta) \) and likelihood \( f(x|\theta) \)
- Procedure:
  - Initialize: Set initial value \( \theta_0 \)
  - Sample: For \( k=1, \ldots, N \)
    - Simulate trial value \( \theta_{\text{trial}} \) from uniform distribution on \( (\theta_{k-1} - \delta, \Theta_{k-1} + \delta) \) for step size \( \delta \)
    - Calculate acceptance ratio \( R = \left\{ \frac{f(x | \theta_{\text{trial}})g(\theta_{\text{trial}})}{f(x | \theta_{k-1})g(\theta_{k-1})} \right\} \)
    - If \( R \geq 1 \), accept and set \( \theta_k = \theta_{\text{trial}} \)
    - If \( R < 1 \), reject with probability \( 1-R \): set \( \theta_k = \begin{cases} \theta_{\text{trial}} & \text{with probability } R \\ \theta_{k-1} & \text{with probability } 1-R \end{cases} \)
- The sequence \( (\theta_0, \ldots, \theta_N) \) is a Markov chain with (under mild regularity conditions) a unique stationary distribution \( g(\theta | X=x) \)
- We will study two extensions:
  - The Hastings correction allows sampling from a distribution other than uniform
  - The component-wise sampler samples one random variable at a time in a graphical model
Review: Gibbs Sampler

- **Objective:** estimate \( g(y|x) = g(y_1, y_2, \ldots, y_p|x) \)
- **Assume:** we cannot sample directly from \( g(y|x) \), but we can sample from each of the “full conditional” distributions \( g(y_i|y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \)
- **Gibbs sampling procedure:**
  - **INITIALIZE:** Chose initial parameter values \( y_1^{(0)}, y_2^{(0)}, \ldots, y_p^{(0)} \)
  - **SAMPLE:** For \( k = 1, \ldots, M \)
    - Sample \( y_1^{(m)} \) from \( g(y_1|y_2^{(m-1)}, \ldots, y_p^{(m-1)}, x) \)
    - Sample \( y_2^{(m)} \) from \( g(y_2|y_1^{(m)}, y_3^{(m-1)}, \ldots, y_p^{(m-1)}, x) \)
    - \( \ldots \)
    - Sample \( y_i^{(m)} \) from \( g(y_i|y_1^{(m)}, \ldots, y_{i-1}^{(m)}, y_{i+1}^{(m-1)}, \ldots, y_p^{(m-1)}, x) \)
    - \( \ldots \)
    - Sample \( y_p^{(m)} \) from \( g(y_p|y_1^{(m)}, \ldots, y_{p-1}^{(m)}, x) \)
- **Result is a Markov chain:** distribution of \( y_1^{(m)}, \ldots, y_p^{(m)} \) is independent of the past given \( y_1^{(m-1)}, \ldots, y_p^{(m-1)} \)
- **Under fairly general conditions** \( g(y|x) \) is the unique stationary distribution of this Markov chain
Component-Wise Metropolis-Hastings Sampler

• For many inference problems:
  – \( g(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \) is expensive or impossible to compute
  – We know \( g(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \) up to a normalizing constant
  – Therefore, \( g(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) / g(y_i' \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \) is easy to compute

• Component-wise Metropolis-Hastings works like Gibbs sampling except that we use a different sampling distribution

• This distribution is chosen to achieve the correct stationary distribution
  – Sample a proposed new value not from \( g(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \) but from some other tractable proposal distribution \( q(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p, x) \)
  – Either accept the newly sampled value as our new value or keep the old value
  – Decision to keep or reject is made probabilistically
  – Acceptance probability is chosen to achieve the correct stationary distribution

• Gibbs sampler is a special case of component-wise MH sampler
Component-Wise MH Procedure

- **Objective:** estimate $g(y|x) = g(y_1, y_2, \ldots, y_p|x)$
- **MH sampling procedure:**
  - **INITIALIZE:** Chose initial parameter values $y_1^{(0)}, y_2^{(0)}, \ldots, y_p^{(0)}$
  - **SAMPLE:** For $m = 1, \ldots, M$
    - For $r = 1, \ldots, p$
      - Sample **trial value** $y^\#$ from **proposal distribution** $q(y | y_1^{(m)}, \ldots, y_p^{(m)}, x)$
      - Compute **acceptance probability**
        $$A(y^\# | y_r^{(m-1)}, y_{-r}^{(m)}, x) = \min \left\{ 1, \frac{q(y_r^{(m-1)} | y^\# y_{-r}^{(m)}, x) \cdot g(y^\# y_{-r}^{(m)} | x)}{q(y_r^{(m-1)} | y_{-r}^{(m)}, x) \cdot g(y_r^{(m-1)} | y_{-r}^{(m)}, x)} \right\}$$
        where $y_{-r}^{(m)} = (y_1^{(m)}, \ldots, y_{r-1}^{(m)}, y_{r+1}^{(m-1)}, \ldots, y_p^{(m-1)})$
      - **Accept** $y^\#$ with probability $A(y^\# | y_r^{(m-1)}, y_{-r}^{(m)}, x)$ and set $y_r^{(m)} = y^\#$
      - If not accepted, **reject** $y^\#$ and keep $y_r^{(m)} = y_r^{(m-1)}$
- **Result is a Markov chain:** distribution of $y_1^{(m)}, \ldots, y_p^{(m)}$ is independent of the past given $y_1^{(m-1)}, \ldots, y_p^{(m-1)}$
- **Under fairly general conditions** $g(y|x)$ is the unique stationary distribution of this Markov chain
MH Acceptance Probability

• The probability of accepting the proposed new state is given by:

$$A(y^\# | y_r^{(m-1)}, y_{r-1}, x) = \min \left\{ 1, \frac{q(y_r^{(m-1)} | y^\# , y_r^{(m)}, x)}{q(y^{(m)} | y_r^{(m-1)}, y_r^{(m)}, x)} \frac{g(y^\# , y_r^{(m)} | x)}{g(y^{(m-1)} , y_r^{(m)} | x)} \right\}$$

• The acceptance ratio consists of two factors:

$$\frac{g(y_1^{(m)}, \ldots, y_r^{(m)}, y^\#, y_{r+1}^{(m-1)}, \ldots, y_p^{(m-1)} | x)}{g(y_1^{(m)}, \ldots, y_{r-1}^{(m)}, y_r^{(m-1)}, y_{r+1}^{(m-1)}, \ldots, y_p^{(m-1)} | x)}$$

$$\frac{h(y_r^{(m-1)} | y_1^{(m)}, \ldots, y_{r-1}^{(m)}, y_r^{(m-1)}, y_{r+1}^{(m-1)}, \ldots, y_p^{(m-1)}, x)}{h(y^\# | y_1^{(m)}, \ldots, y_{r-1}^{(m)}, y_r^{(m-1)}, y_{r+1}^{(m-1)}, \ldots, y_p^{(m-1)}, x)}$$

• Acceptance probability is larger if:
  – Posterior probability of proposed new state is higher
  – Transition back to old state is more likely

• Transition probability satisfies detailed balance

$$q(y^\# | y_r, y_{r-1}, x) A(y^\# | y_r, y_{r-1}, x) g(y_r, y_{r-1} | x)$$

$$= q(y_r | y^\#, y_{r-1}, x) A(y_r | y^\#, y_{r-1}, x) g(y^\#, y_{r-1} | x)$$

At equilibrium distribution $g(\cdot)$, transitions from $y_r$ to $y^\#$ balance transitions from $y^\#$ to $y_r$
Comments on MH Sampler and Gibbs Sampler

• The Metropolis-Hastings sampler has the same stationary distribution as the Gibbs sampler

• Metropolis-Hastings sampler is same as Gibbs sampler when we sample from the full conditional distribution:

\[ q(y \mid y_1^{(m)}, \ldots, y_p^{(m)}, x) = g(y \mid y_1^{(m)}, \ldots, y_{r-1}^{(m)}, y_{r+1}^{(m)}, \ldots, y_p^{(m)}, x) \]

In this case the acceptance probability is 1

• MH acceptance probability is higher when:
  – Posterior probability of new state is higher
  – Transition back to old state is more likely

• Under regularity conditions, the MH stationary distribution is unique and satisfies a central limit theorem
  – e.g., proposal distribution \( q(\cdot) \) and acceptance probability \( A(\cdot) \) do not change as sampling progresses and all values of each \( y_r \) have strictly positive probability density for all values of \( y_1^{(m)}, \ldots, y_p^{(m)}, x \)

• Both Gibbs and component-wise MH suffer from the problem of getting stuck in local basins of attraction

• Thinning is especially important for MH sampling to avoid successive samples having identical values
Importance Sampling

- We want to estimate $\mathbb{E}[h(X)]$, where $X \sim f(x)$
- Direct Monte Carlo would sample from $f(x)$:
  - Sample $X_1, \ldots, X_N$ from $f(x)$
  - Use estimator $\hat{E}[h(X)] \approx \frac{1}{N} \sum_{n=1}^{N} h(X_n)$
- In some problems, we cannot sample from $f(x)$, but for any given value $X=x$, we can calculate a value $a(x) \propto f(x)$ proportional to $f(x)$
- In this case, we can sample from another distribution $q(x)$, and use weights to correct the estimate
  - $q(x)$ is called the importance distribution
  - The weights are called importance weights
- Importance sampling algorithm:
  - For $n = 1, \ldots, N$
    - Sample $X_n$ from importance distribution $q(x)$
    - Calculate $w_n = \frac{a(X_n)}{q(X_n)}$ where $a(X_n) = zf(x_n)$ is proportional to $f(x_n)$
  - The importance sampling estimate is: $\hat{E}[h(X)] \approx \frac{1}{N} \sum_{n=1}^{N} \frac{w_n h(X_n)}{N} = \sum_{n=1}^{N} \left( \frac{w_n}{\sum_{n=1}^{N} w_n} \right) h(X_n)$
Why Importance Sampling Works

• Our goal is to estimate $E_f[h(X)] = \int h(x)f(x)d\mu(x)$

• If we sampled from $q(x)$ and used the naïve direct Monte Carlo estimate, the expected value would be $E_q[h(X)] = \int h(x)q(x)d\mu(x)$, which is not the value we want.

• Importance weights adjust to correct the expectation.

• Assume we can calculate a value $a(x) = zf(x)$ proportional to $f(x)$.

• Then:

  $- E_q[W] = E_q\left[\frac{a(X)}{q(X)}\right] = \int x\left(\frac{a(x)}{q(x)}\right)q(x)d\mu(x) = \int a(x)d\mu(x) = z$

  $- E_q[Wh(X)] = E_q\left[\frac{a(X)}{q(X)}h(X)\right] = \int x\left(\frac{a(x)}{q(x)}\right)h(X)q(x)d\mu(x) = \int h(x)a(x)d\mu(x) = zE_f[h(x)]$

• Therefore by the Law of Large Numbers:

  $- \sum_{k=1}^{N} w_k \xrightarrow{N\to\infty} z$ and $\sum_{n=1}^{N} w_n h(X_n) \xrightarrow{N\to\infty} zE_f[h(X)]$

• Therefore, the importance sampling estimate $\frac{\sum_{n=1}^{N} w_n h(X_n)}{\sum_{n=1}^{N} w_n}$ converges to $zE_f[h(X)]$ as $N$ tends to infinity.
More on Importance Sampling

- \( \left( \sum_{k=1}^{N} w_k, \sum_{n=1}^{N} w_n h(X_n) \right) \) satisfies a central limit theorem
- A good importance distribution is crucial
- Variance of estimate is smallest when the importance distribution \( g(x) \) is near \( f(x) \)
  - If \( g(x) = f(x) \), then importance sampling is just direct Monte Carlo
  - Variance may be very high if there are values \( x \) for which \( g(x) \) is much larger than \( f(x) \)
- Adaptive importance sampling:
  - Adapt importance distribution as sampling progresses
  - Has shown good results
  - Must be careful not to adapt too quickly
Importance Sampling for Posterior Quantities

• Suppose we want to estimate a posterior expected value

\[ E_{f(\theta|x)}[h(\Theta | X = x)] = \int h(\theta) f(\theta | x) d\mu(\theta) \]

• In many cases, direct Monte Carlo sampling from the posterior distribution is not possible

• We can use importance sampling to compute posterior expectations
  – Note that \( a(\theta) = f(x | \theta)g(\theta) = f(x)g(\theta | x) \) is proportional to the posterior density function
  – For \( n = 1, \ldots, N \)
    » Sample \( \Theta_n \) from importance distribution \( q(\theta) \)
    » Calculate \( w_n = \frac{a(\theta_n)}{q(\theta_n)} = \frac{f(x | \theta_n)g(\theta_n)}{q(\theta_n)} \) where \( g(\theta_n) \) is the prior density and \( f(x | \theta_n) \)
      is the likelihood
  – The importance sampling estimate is:

\[ E[h(\Theta | X = x)] \approx \sum_{n=1}^{N} \frac{w_n h(\theta_n)}{\sum_{n=1}^{N} w_n} = \sum_{n=1}^{N} \left( \frac{w_n}{\sum_{n=1}^{N} w_n} \right) h(\theta_n) \]
Forward Importance Sampling in Graphical Models

Forward importance sampling is a commonly used approximate inference method in graphical models.

- For \( n = 1, \ldots, N \)
  - Traverse the graph in the direction of the arcs, beginning at a root and sampling each unobserved random variable only after its parents have been sampled.
  - Observed variables remain clamped at their known values.
  - Sample each unobserved variable \( X_i^{(n)} \), \( i = 1, \ldots, p \) from an importance distribution \( q_i(x_i | x_{pa(i)}^{(n)}) \), where \( X_{pa(i)} \) are the parents of \( X_i \) in the graph.
  - Calculate importance weight \( w_n \) for sampled value \((X_1^{(n)}, \ldots, X_p^{(n)})\).

- Construct importance sampling estimate for quantities of interest.

For rat tumors example:

- For \( n = 1, \ldots, N \)
  - Sample \( U^{(n)} \) from \( q_u(u) \)
  - Sample \( V^{(n)} \) from \( q_u(v|U^{(n)}) \)
  - For each \( i \), sample \( \Theta_i^{(n)} \) from \( q(\Theta_i | U^{(n)} , V^{(n)}) \)
  - Calculate \( w_n = \frac{f(Y | \Theta_i^{(n)}) g(\Theta_i^{(n)} | U^{(n)}, V^{(n)}) g(U^{(n)}, V^{(n)})}{q(\Theta_i^{(n)} | U^{(n)}, V^{(n)}) g(U^{(n)}, V^{(n)})} \)
  - Use the sampled values \((U_n, V_n, \Theta_n)\) and weights \( w_n \) to compute posterior estimates.
Sampling – Importance Resampling

- Importance sampling results in a set of random draws with weights, \((x_n, w_n)\), \(n=1,\ldots,N\), where the weights \(w^{(n)}\) are chosen so that expectations of weighted sums match expectations under target distribution \(f(x)\).

- Many statistical analysis methods assume observations are an unweighted iid sample from some distribution.

- Therefore, it may be impossible to analyze an importance sample using standard statistical software.

- Sampling – Importance Resampling converts the importance sample to an unweighted sample that approximates our target distribution \(f(x)\).

- Sampling – Importance Resampling method:
  - Step 1: Sample \(\{x_1, \ldots, x_N\}\) as iid draws from importance distribution \(q(x)\).
  - Step 2: Calculate importance weights \((w_1, \ldots, w_N)\), where \(w_i = zf(x_i)/q(x_i)\).
  - Step 3: Resample \(\{x'_1, \ldots, x'_m\}\) as iid draws with replacement from \(\{x_1, \ldots, x_N\}\), where \(x_i\) is chosen with probability \(p_i = w_i/(\Sigma_j w_j)\) proportional to the importance weight.

- To prevent too many repeated values, the size \(m\) of the final sample should be smaller than the size \(N\) of the original sample.
  - A common rule of thumb is \(m < 0.1N\).
Comparison: Forward Sampling and MH/Gibbs Sampling

• Forward sampling with importance weights
  – Sample unobserved variables in direction of graph
    \( X_i \) is sampled after \( X_{pa(i)} \)
  – Each sample iteration \( X^{(n)} \) is an independent draw from from importance distribution \( q(\cdot) \)
  – Correct expectation is achieved by giving each draw an importance weight proportional to ratio of target density and importance density

• MH/Gibbs Sampling
  – Begin by assigning arbitrary values to random variables
  – Visit random variables in any order
    » Propose new value for variable using distribution that depends on current value of other random variables
    » Accept new state with probability depending on relative probabilities of new and old states, and relative proposal likelihoods of new and old states
  – Sampling distribution is a Markov chain with unique stationary distribution equal to target distribution
Sequential Monte Carlo

• Sequential Monte Carlo (SMC) refers to a class of Monte Carlo methods for estimating posterior distributions for systems that evolve in time

• SMC estimators are also called particle filter methods
  – A set of “particles” (Monte Carlo draws) is used to estimate a probability distribution for the system state at a given time
  – Stochastic state transitions are estimated by Monte Carlo simulation of the next state of each particle given its previous state

• Commonly used particle filters make use of methods we have discussed
  – Particle weights (importance weights) reflect relative contribution of each particle to the estimated state distribution
  – Resampling is used to keep importance weights from becoming to unevenly distributed
  – Metropolis-Hastings sampling can be combined with forward sampling to mitigate “particle collapse” (concentration of particles at a local mode of the likelihood)
State Space Model

- A state space model is a representation for a dynamic system that satisfies the following conditions:
  - The behavior of the system depends on a state $X_t$ which evolves in time
  - The state $X_t$ at time $t$ depends on the past only through its dependence on the immediate past state $X_{t-1}$ (Markov assumption)
    
    \[ P(X_t \mid X_0, X_1, \ldots, X_{t-1}) = P(X_t \mid X_{t-1}) \]
  - The state is hidden and cannot be observed directly
  - We learn about the state through an observation $Y_t$ which depends on $X_t$ but not on past states or past observations
    
    \[ P(Y_t \mid X_0, X_1, \ldots, X_t, Y_0, Y_1, \ldots, Y_{t-1}) = P(Y_t \mid X_t) \]

- The state may be a complex multivariate quantity
- State space models are a powerful and general way to model systems that evolve in time
Typical Tasks for State Space Models

• Prediction -
  – Predict states of future variables given observations on past variables

• Filtering -
  – Infer values of current unobserved variables given current and past observed variables

• Smoothing -
  – Revise inferences about past unobserved variables given new observations

• Estimation -
  – Infer hidden static parameters from observations

These tasks may be performed offline in batch mode or online sequentially
Examples

- Hidden Markov model
  - State is discrete, no internal structure
  - Observation is discrete
  - State transition is discrete Markov chain
  - Many applications in pattern recognition: speech recognition, language understanding, image understanding…

- Kalman filter
  - State is multivariate Gaussian
  - Observation is multivariate Gaussian
  - State transition is linear
  - Many applications in tracking and filtering

- Dynamic Bayesian network
  - State is usually discrete and multivariate
  - Observation is usually discrete, can be multivariate
  - Factored state transition probability distribution is represented as graph plus local distributions
  - Many applications in robotics, artificial intelligence, multi-source fusion

- Partially dynamic Bayesian network
  - State includes one or more static (not changing with time) variables

All these examples are special cases of partially dynamic Bayesian networks
Review: Kalman Filter

- **Goal:** follow a moving object with unknown position and velocity
- **The algorithm:**
  1. Get observation (noisy) on object position
  2. Use Bayesian inference to refine estimate of position and velocity
  3. Use estimate of position and velocity to make (noisy) prediction of position and velocity at time of next observation
  4. Return to Step 1
- There are closed form equations for online filtering, prediction, estimation using Kalman filter
- There are many extensions to deal with nonlinearities and other relaxations of assumptions of basic model
General Filtering

- Kalman filter is based on assumption:
  - State and observation are normally distributed
  - Evolution equation is linear
- There are extensions to the basic Kalman filter to handle nonlinearities and non-normal distributions
- For some problems the extensions are insufficient
- Monte Carlo algorithms are general-purpose approximation methods that make few distributional assumptions
- Sequential Monte Carlo uses the Bayesian recursion equations to perform online Monte Carlo filtering
Sequential Monte Carlo (Particle Filter)

Initialization

Importance sampling

Resampling

Evolution

Importance sampling

i=1,...,N=10 particles

\{\tilde{x}_{t-1}^{(i)}, N^{-1}\}

\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}

\{x_t^{(i)}, N^{-1}\}

\{x_{t-1}^{(i)}, N^{-1}\}

\{\tilde{x}_t^{(i)}, N^{-1}\}

\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}

(graphics taken from Van der Merwe et al.)
Particle Filter Algorithm

• The problem:
  – State space model with state $X_t$ and observation $Y_t$ at time $t$
    » State and observation can be univariate or multivariate, continuous or discrete
  – We are given an initial distribution $f_0(x)$, a transition distribution $f(x_t \mid x_{t-1})$, and an observation distribution $g(y_t \mid x_t)$

• Initialization:
  – Sample initial particles $\{x_0^{(1)}, \ldots, x_0^{(N)}\}$ (exactly or approximately) from $f_0(x)$
  – Set initial weights $\{w_0^{(1)}, \ldots, w_0^{(N)}\} = \{N^{-1}, \ldots, N^{-1}\}$ the same for all particles

• For $t=1, \ldots, T$
  – For each particle $x_{t-1}^{(k)}$, $k=1,\ldots,N$
    » Sample new trial particle $\tilde{x}_t^{(k)}$ from $f(x_t \mid x_{t-1}^{(k)})$
    » Calculate weight $\tilde{w}_t^{(k)} = w_{t-1}^{(k)} g(y_t \mid \tilde{x}_t^{(k)})$
  – Calculate normalized weights $p_t^{(k)} = \tilde{w}_t^{(k)} / \sum_i \tilde{w}_t^{(i)}$
  – If weights have become too uneven then resample
    » Sample $N$ new particles $\{x_t^{(1)}, \ldots, x_t^{(N)}\}$ with replacement from trial particles $\{\tilde{x}_t^{(1)}, \ldots, \tilde{x}_t^{(N)}\}$ with probabilities $\{p_t^{(1)}, \ldots, p_t^{(N)}\}$
    » Reset to equal weights $\{w_t^{(1)}, \ldots, w_t^{(N)}\} = \{N^{-1}, \ldots, N^{-1}\}$
  – Else set $\{x_t^{(1)}, \ldots, x_t^{(N)}\} = \{\tilde{x}_t^{(1)}, \ldots, \tilde{x}_t^{(N)}\}$ and $\{w_t^{(1)}, \ldots, w_t^{(N)}\} = \{p_t^{(1)}, \ldots, p_t^{(N)}\}$
Visualizations

- Animations of particle filters for robotics:

Rao-Blackwellized particle filter for object tracking
- Red circles are particles for robot location
- White circles are Kalman filters for the ball’s location and velocity
- Colored circles are landmark detections
Particle Filter: Limiting Behavior

• Under reasonable conditions on the importance distribution, importance sampling estimates satisfy a central limit theorem.

• As number of samples becomes large:
  – Estimate converges with probability 1 to the integral being estimated.
  – Deviations of estimate from true value have approximately a Gaussian distribution.

• These results apply as number of particles becomes large while holding the number of time steps fixed.

• We are often interested in holding number of samples fixed and letting number of time steps become large.

• There are no large-sample results for this case!
Modifications of Basic Particle Filter

- Importance sampling
  - Sample trial value $x'_t$ from distribution other than $P(x_t | x_{t-1})$ and weight the estimate appropriately
  - The optimal sampling distribution would be $P(x_t | x_{t-1}, y)$ but this may be intractable
  - Poorly chosen importance distribution can make things worse
  - Adaptive importance sampling repeats importance sampling, improving the importance distribution on each step
  - It is important not to adapt too aggressively to avoid overfitting

- Rao-Blackwellization
  - Sometimes we can break a multivariate state into tractable sub-parts by sampling only some of the variables and doing exact computation on the rest
  - Theory: replacing sampled value by expectation lowers variance of estimator (Rao-Blackwell theorem)
Challenge: Particle Impoverishment

• After several rounds of resampling, typically all particles are descended from a single initial particle
• When there are static variables or near-deterministic transitions this can cause very poor performance
• Particle impoverishment can result in convergence to local minima of likelihood function and very poor estimates
• There is no asymptotic theory for particle filter as number of timesteps becomes large -- asymptotic theory relates to number of particles becoming large
• There are no guarantees of convergence
Dealing with Impoverishment

- More particles (brute force)
  - Usually not very effective when particle impoverishment is severe (especially in case of static variables)

- Regularized particle filter
  - Ordinary particle filter uses discrete approximation to state density (the set of sample points)
  - Regularized particle filter
    - Approximate state density at past time step with continuous distribution
    - Often use mixture of Gaussians with small standard deviation (kernel density estimator; standard deviation is bandwidth)
    - Resample from approximate density before propagating to next time step

- Adaptive importance sampling
  - A good importance distribution is the best solution to particle impoverishment
  - Bad importance distribution can make impoverishment much worse
  - Adaptive importance sampling iteratively improves approximation
  - Computation cost is worth it if good importance distribution is found
Particle Impoverishment with Static Parameters

• Standard PF *cannot recover* from impoverishment of static parameter

• Suggested approaches:
  – Artificial evolution of static parameter
    » Ad hoc; no justification for amount of perturbation; information loss over time
  – Shrinkage (Liu & West)
    » Combines ideas from artificial evolution & kernel smoothing
    » Perturbation “shrinks” static parameter for each particle toward weighted sample mean
      • Perturbation holds variance of set of particles constant
      • Correlation in disturbances compensates for information loss
  – Resample-Move (Gilks & Berzuini)
    » Metropolis-Hastings step corrects for particle impoverishment
    » MH sampling of static parameter involves entire trajectory but is performed less frequently as runs become longer

• There is not much literature on empirical performance of these approaches in applications
Summary of Key Ideas in Monte Carlo Sampling

- **Rejection sampling**
  - Sample observations from a distribution
  - Reject observations according to criterion
  - Estimate target quantity from remaining observations
  - Resulting estimate approximates target quantity
  - Examples: Logic sampling and Metropolis-Hastings

- **Importance sampling**
  - Sample observations from a distribution
  - Compute sampling weight
  - Estimate target quantity as ratio of weighted sums
  - Resulting estimate approximates target quantity
  - Likelihood weighting can be viewed as importance sampling with the importance distribution equal to the prior distribution

- **“Rao Blackwellization”**
  - Replace the sampled quantity with its expectation when the expectation can be computed cheaply
  - Examples: likelihood weighting and Markov blanket scoring

- **Markov Chain sampling**
  - Sometimes a distribution is difficult to compute or sample directly but is the stationary distribution of a Markov chain that is cheap to sample
  - Sampling from the Markov chain approximates target distribution
  - Examples: Gibbs sampling and Metropolis-Hastings sampling
References

General Object Tracking


Particle filters and related Monte Carlo methods