Reasoning Over Time
[RN2] Sec 15.1-15.3, 15.5
[RN3] Sec 15.1-15.3, 15.5

CS 486/686

University of Waterloo

Lecture 14: February 27, 2014

Sampling Techniques

- Direct sampling
- Rejection sampling
- Likelihood weighting
- Importance sampling
- Particle Filtering (a.k.a. sequential Monte Carlo sampling)

Approximate Inference by Sampling

- Expectation: $E_P[f(x)] = \int_x P(x)f(x)dx$
  - Approximate integral by sampling: $E_P[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$ where $x_i \sim P(x)$
- Inference query: $\Pr(\boldsymbol{X}|e) = \sum_{\boldsymbol{Y}} \Pr(\boldsymbol{X}, \boldsymbol{Y}|e)$
  - Approximate exponentially large sum by sampling: $\Pr(\boldsymbol{X}|e) = \frac{1}{n}\sum_{i=1}^{n} \Pr(\boldsymbol{X}|\boldsymbol{y}_i, e)$ where $\boldsymbol{y}_i \sim P(\boldsymbol{Y}|e)$

Direct Sampling (a.k.a. forward sampling)

- Unconditional inference queries (i.e., $\Pr(V = t)$)
- Bayesian networks
  - Idea: sample each variable given the values of its parents according to the topological order of the graph.

Direct Sampling Algorithm

Sort the variables by topological order

For $i = 1$ to n do (sample n particles)

For each variable V_j do

Sample $v_j^{(i)} \sim \Pr(V|\boldsymbol{pa}_V)$

- Approximation: $\Pr(V_k = t) \approx \frac{1}{n}\sum_{i=1}^{n} \delta(v_k^{(i)} = t)$

Example

Analysis

- Complexity: $O(n|\boldsymbol{V}|)$ where $|\boldsymbol{V}| = \#\text{variables}$
- Accuracy
  - Absolute error ϵ: $\mathrm{P}\left(\left|\widehat{\mathrm{P}}(\mathrm{V}) - \mathrm{P}(\mathrm{V})\right| > \epsilon\right) \leq \delta = 2e^{-2n\epsilon^2}$
    - Sample size $n \geq \frac{\ln\left(\frac{2}{\delta}\right)}{2\epsilon^2}$
  - Relative error ϵ: $P\left(\frac{\hat{P}(V)}{P(V)} \notin [1-\epsilon, 1+\epsilon]\right) \leq \delta = 2e^{-\frac{nP(V)\epsilon^2}{3}}$
    - Sample size $n \geq \frac{3\ln\left(\frac{2}{\delta}\right)}{2P(V)\epsilon^2}$

Rejection Sampling

- Conditional inference queries (i.e., $\Pr(V = t|\boldsymbol{e})$)
- Bayesian networks
  - Idea: sample each variable given the values of its parents according to the topological order of the graph, however reject samples that do not agree with evidence

Rejection Sampling Algorithm

Sort the variables by topological order

For $i = 1$ to n do (sample n particles)

For each variable V_j do

Sample $v_j^{(i)} \sim \Pr(V|\boldsymbol{pa}_V)$

Reject $\boldsymbol{v}^{(i)}$ if $\boldsymbol{v}^{(i)}$ is inconsistent with $\boldsymbol{e}$ (i.e., $\boldsymbol{v}_{\boldsymbol{E}}^{(i)} \neq \boldsymbol{e}$)

- Approximation: $\Pr(V_k = t|\boldsymbol{e}) \approx \frac{\sum_{i=1}^{n} \delta\left(v_k^{(i)}=t \wedge \boldsymbol{v}_{\boldsymbol{E}}^{(i)}=\boldsymbol{e}\right)}{\sum_{i=1}^{n} \delta\left(\boldsymbol{v}_{\boldsymbol{E}}^{(i)}=\boldsymbol{e}\right)}$

Example

Analysis

- Complexity: $O(n|\boldsymbol{V}|)$ where $|\boldsymbol{V}| = \#\text{variables}$
- Expected # samples that are accepted: $O(n\,\Pr(\boldsymbol{e}))$
  - Since $\Pr(\boldsymbol{e})$ often decreases exponentially with the number of evidence variables, the number of samples also decreases exponentially.
  - For good accuracy: exponential # of samples often needed in practice.

Likelihood Weighting

- Conditional inference queries (i.e., $\Pr(V = t|\boldsymbol{e})$)
- Bayesian networks
  - Idea: sample each non-evidence variable given the values of its parents in topological order. Assign weights to samples based on the probability of the evidence.

Likelihood Weighting Algorithm

Sort the variables by topological order
For $i = 1$ to n do (sample n particles)
    $w_i \leftarrow 1$
    For each variable V_j do
        If V_j is not an evidence variable do
            Sample $v_j^{(i)} \sim \Pr(V_j | \boldsymbol{pa}_V)$
        else
            $w_i \leftarrow w_i * \Pr\left(v_j \middle| \boldsymbol{pa}_{V_j}\right)$

- Approximation: $\Pr(V_k = t|\boldsymbol{e}) \approx \frac{\sum_{i=1}^{n} w_i \delta\left(v_k^{(i)}=t\right)}{\sum_{i=1}^{n} w_i}$

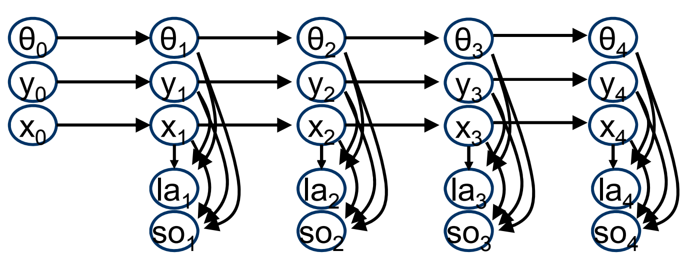
Example

Analysis

- Complexity: $O(n|\boldsymbol{V}|)$ where $|\boldsymbol{V}| = \#\text{variables}$
- Effective sample size: $O(n\,\Pr(\boldsymbol{e}))$
  - Even though all samples are accepted, their importance is reweighted to a fraction equal to $\Pr(\boldsymbol{e})$
  - For good accuracy: the # of samples will be the same as for rejection sampling (hence exponential with the number of evidence variables).

Particle Filtering

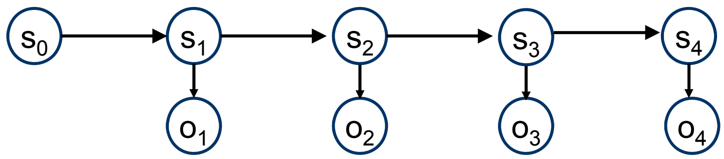
- Variant of likelihood weighting for sequential processes such as dynamic Bayesian networks



- # of particles needed grows exponentially with the number of observations.

Sequential Inference

- Consider an HMM with continuous states

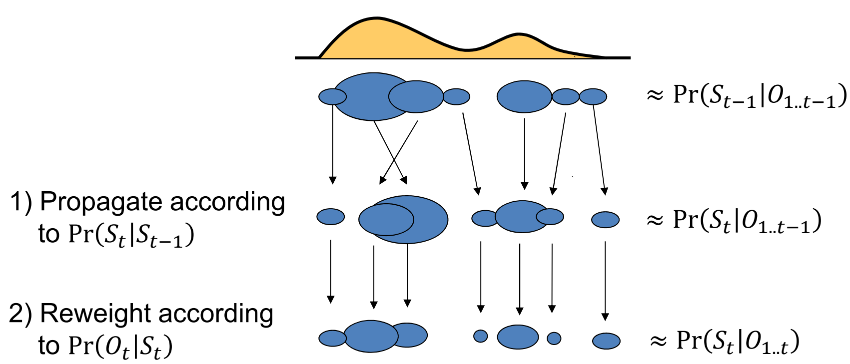


- Consider the following sequence of queries: $\Pr(S_1|O_1), \Pr(S_2|O_{1..2}),\ \Pr(S_3|O_{1..3})$, etc.
- Variable elimination can answer these queries incrementally. This leads to a recursive equation:

$$\Pr(S_t|O_{1..t}) = \sum_{S_{t-1}} \Pr(S_{t-1}|O_{1..t-1}) \Pr(S_t|S_{t-1}) \Pr(O_t|S_t)$$

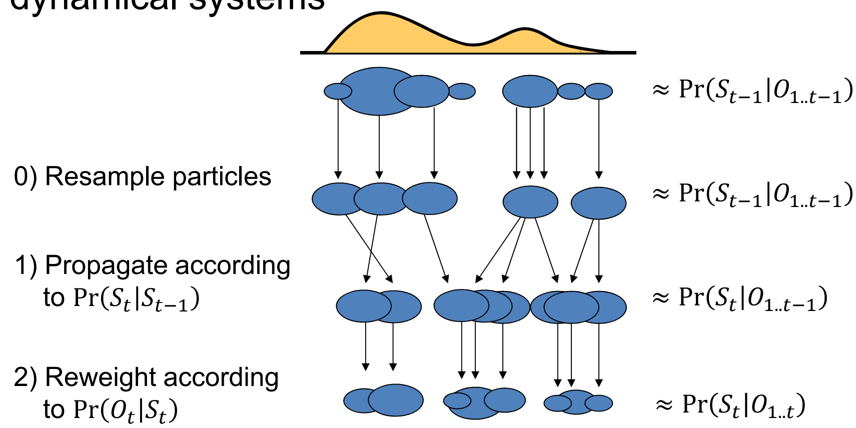
Recursive Likelihood weighting

$$\Pr(S_t|O_{1..t}) = \sum_{S_{t-1}} \Pr(S_{t-1}|O_{1..t-1}) \Pr(S_t|S_{t-1}) \Pr(O_t|S_t)$$



Particle Filtering

- Likelihood weighting with resampling for dynamical systems



Particle Filtering Algorithm

Sample n particles $s_1^0, \dots, s_n^0$ from $\Pr(S_0)$

Initialize the particle weights $w_1, \dots, w_n$ to 1

For $t = 1$ to $horizon$ do

Resample n particles according to the distribution implied by the weights and assign a weight of 1 to each particle.

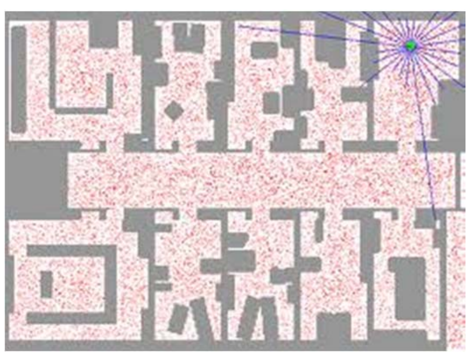
Sample new particles $s_i^t \sim \Pr\left(S_t \middle| s_i^{t-1}\right) \quad \forall i$

Reweight each particle: $w_i \leftarrow w_i * \Pr\left(o_t \middle| s_i^t\right) \quad \forall i$

Analysis

- The number of particles needed for good accuracy depends on the length of the "effective history" of the process
- The effective history is the set of past observations that is sufficient to determine the next state
- In many applications, processes are forgetful in the sense that only a handful of recent observations tend to matter.

Robot localisation



- University of Washington robotics and State Estimation
- http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/

Neato Robotics

- Robotic Vacuum Cleaners by Neato Robotics

- Use particle filtering for simultaneous localisation and mapping
- See patent:

  http://www.faqs.org/patents/assignee/neato-robotics-inc/