CS786
Lecture 13: May 14, 2012

Sampling techniques

[KF Chapter 12]

Sampling Techniques

- Direct sampling
- Rejection sampling
- Likelihood weighting
- Importance sampling
- Markov chain Monte Carlo (MCMC)
  - Gibbs Sampling
  - Metropolis-Hastings
- Sequential Monte Carlo sampling (a.k.a. particle filtering)

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 only
  - 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 only
  - 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 only
  - 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).

Importance Sampling

- Likelihood weighting is a special case of importance sampling
- General approach to estimate $E_P[f(x)]$ by sampling from Q instead of P
  - Works for Bayes nets and probability densities
- Idea: generate samples x from Q and assign weights $P(x)/Q(x)$

Importance Sampling Algorithm

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

Sample $x^{(i)}$ from Q

Assign weight: $w_i \leftarrow P(x^{(i)})/Q(x^{(i)})$

- Approximation: $E_P[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} w_i f\left(x^{(i)}\right)$
  - Unbiased estimator
  - Variance of estimator decreases linearly with sample size

Normalized Importance Sampling

- Often the reason why we are sampling from Q instead of P is that we don't know P.
- But, we may know $\tilde{P}$ an unnormalized version of P
  - Markov nets: $P(\boldsymbol{X}) = k \prod_i f_i(\boldsymbol{X})$ while $\tilde{P}(\boldsymbol{X}) = \prod_i f_i(\boldsymbol{X})$
  - Bayes nets: $P(\boldsymbol{X}|\boldsymbol{e})$ while $\tilde{P}(\boldsymbol{X}, \boldsymbol{e})$

- Idea: generate samples x from Q and assign weights $\tilde{P}(x)/Q(x)$. Normalize the estimator.

Normalized Importance Sampling Algorithm

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

Sample $x^{(i)}$ from Q

Assign weight: $w_i \leftarrow \tilde{P}(x^{(i)})/Q(x^{(i)})$

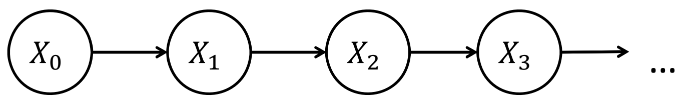
- Approximation: $E_P[f(x)] \approx \frac{\sum_{i=1}^{n} w_i f(x^{(i)})}{\sum_{i=1}^{n} w_i}$
  - Biased estimator for finite n (unbiased for $n = \infty$)
  - Variance of estimator decreases linearly with sample size

Markov Chain Monte Carlo

- Iterative sampling technique that converges to the desired distribution in the limit
- Idea: set up a Markov chain such that its stationary distribution is the desired distribution

Markov Chain

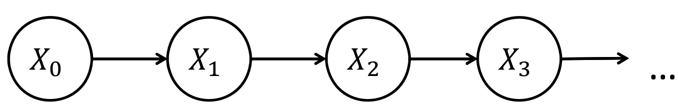
- Definition: A Markov chain is a linear chain Bayesian network with a stationary conditional distribution known as the transition function



- Initial distribution: $\Pr(X_0)$
- Transition distribution: $\Pr(X_t | X_{t-1})$

Markov Chain

- Definition: A Markov chain is a linear chain Bayesian network with a stationary conditional distribution known as the transition function



- Initial distribution: $\Pr(X_0)$
- Transition distribution: $\Pr(X_t | X_{t-1})$

Asymptotic Behaviour

- Let $\Pr(X_t)$ be the distribution at time step t

$$\Pr(X_t) = \sum_{\boldsymbol{X}_{0..t-1}} \Pr(\boldsymbol{X}_{0..t})$$
$$= \sum_{X_{t-1}} \Pr(X_{t-1}) \Pr(X_t | X_{t-1})$$

- In the limit (i.e., when $t \to \infty$), the Markov chain may converge to stationary distribution $\pi(x) = \Pr(X_\infty = x)$

$$\pi(x) = \Pr(X_\infty = x)$$
$$= \sum_{X_{\infty-1}} \Pr(X_{\infty-1} = x') \Pr(X_\infty = x | X_{\infty-1} = x')$$
$$= \sum_{x'} \pi(x') \Pr(x|x')$$

Stationary distribution

- Let $T_{x|x'} = \Pr(x|x')$ be a matrix that represents the transition function

- If we think of π as a column vector, then π is an eigenvector of T with eigenvalue 1

$$T\pi = \pi$$

Ergodic Markov Chain

- Definition: A Markov chain is ergodic when there is a non-zero probability of reaching any state from any state in a finite number of steps
- When the Markov chain is ergodic, there is a unique stationary distribution

- Sufficient condition: detailed balance

$$\pi(x)\Pr(x'|x) = \pi(x')\Pr(x|x')$$

Detailed balance → ergodicity → unique stationary dist.

Markov Chain Monte Carlo

- Idea: set up an ergodic Markov chain such that the unique stationary distribution is the desired distribution

- Since the Markov chain is a linear chain Bayes net, we can use direct sampling (forward sampling) to obtain a sample of the stationary distribution

Generic MCMC Algorithm

Sample $x_0 \sim \Pr(X_0)$
For $i = 1$ to n do (sample n particles)
  Sample $x_t \sim \Pr(X_t | x_{t-1})$

- Approximation: $\pi(x) \approx \frac{1}{n} \sum_{t=1}^{n} \delta(x_t = x)$
- In practice, ignore the first k samples for a better estimate (burn-in period):

$$\pi(x) \approx \frac{1}{n-k} \sum_{t>k}^{n} \delta(x_t = x)$$

Choosing a Markov Chain

- Different Markov chains lead to different algorithms
  - Gibbs sampling
  - Metropolis Hastings

Gibbs Sampling

- Suppose $\Pr(\boldsymbol{X})$ defined by a graphical model (Bayes net or Markov net)
- Inference query: $\Pr(\boldsymbol{Y}|\boldsymbol{e})$? Where $\boldsymbol{Y} \subseteq \boldsymbol{X}$

- Idea: randomly assign values to all non-evidence variables, then repeatedly sample each non-evidence variable given the assigned values for all other variables

Gibbs Sampling Algorithm

Randomly assign $v_j^{(0)}$ to all non-evidence variables V_j

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

For each non-evidence variable V_j do

Sample $v_j^{(i)} \sim \Pr\left(V_j \middle| \boldsymbol{v}_{\sim j}^{(i-1)}, \boldsymbol{e}\right)$

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

Example

Practical Consideration

- Burn-in period: ignore first k samples:

$$\Pr(V_k = t|\boldsymbol{e}) \approx \frac{1}{n-k}\sum_{i>k}^{n} \delta(v_k^{(i)} = t)$$

- Use most recent values to sample $V_j^{(i)}$

$$v_j^{(i)} \sim \Pr(V_j^{(i)}|\boldsymbol{v}_{1\ldots j-1}^{(i)}, \boldsymbol{v}_{j+1\ldots|\boldsymbol{V}|}^{(i-1)})$$

- Use conditional independence to restrict parent variables to the Markov blanket

$$v_j^{(i)} \sim \Pr(V_j^{(i)}|\boldsymbol{v}_{\forall k<j, k\in mb(j)}^{(i)}, \boldsymbol{v}_{\forall k>j, k\in mb(j)}^{(i-1)})$$

Convergence

- Let $\Pr(\boldsymbol{V}^{(i)}|\boldsymbol{V}^{(i-1)}, \boldsymbol{e})$ be the transition function of the Markov chain associated with Gibbs sampling

- **Theorem:** Gibbs sampling converges to $\Pr(\boldsymbol{V}|\boldsymbol{e})$ when all potentials are strictly positive.
- Proof: $\Pr(\boldsymbol{V}^{(i)}|\boldsymbol{V}^{(i-1)}, \boldsymbol{e})$ satisfies detailed balance
  i.e. $\Pr(\boldsymbol{V}|\boldsymbol{e}) \Pr(\boldsymbol{V}'|\boldsymbol{V}, \boldsymbol{e}) = \Pr(\boldsymbol{V}'|\boldsymbol{e}) \Pr(\boldsymbol{V}|\boldsymbol{V}', \boldsymbol{e})$

Metropolis-Hastings

- Suppose we can compute $\pi(x)$ for a given x but we can't sample from $\pi(x)$ easily.
- Idea: use an arbitrary transition distribution $Q(x'|x)$ and use se rejection sampling to correct for the choice of Q.

- Advantage: since Q can be anything, we can always obtain an MCMC algorithm by Metropolis-Hastings
  - It is particularly useful to approximate continuous distributions

Metropolis-Hastings Algorithm

Randomly select $x^{(0)}$
For $i = 1$ to n do (sample n particles)
Sample $x^{(i)} \sim \mathrm{Q}\left(X \middle| x^{(i-1)}\right)$
Accept $x^{(i)}$ with probability $\min\left[1, \frac{\pi(x^{(i)})Q(x^{(i)}|x^{(i-1)})}{\pi(x^{(i-1)})Q(x^{(i-1)}|x^{(i)})}\right]$
Otherwise reject $x^{(i)}$ (i.e., $x^{(i)} \leftarrow x^{(i-1)}$)

- Approximation: $\pi(x) \approx \frac{1}{n}\sum_{i=1}^{n} \delta(x^{(i)} = x)$

Convergence

- The transition distribution in Metropolis-Hastings is

$$\Pr(x'|x) = \begin{cases} Q(x'|x)A(x \to x') & x \neq x' \\ Q(x|x) + \sum_{x' \neq x} Q(x'|x)(1 - A(x \to x')) & x = x' \end{cases}$$

where $\mathrm{A}(\mathrm{x} \to x') = \min\left[1, \frac{\pi(x')Q(x'|x)}{\pi(x)Q(x|x')}\right]$

- **Theorem:** Metropolis-Hastings converges to $\pi(x)$.
- Proof: $\Pr(x'|x)$ satisfies detailed balance