Sampling Methods
Greg Mori - CMPT 419/726

Bishop PRML Ch. 11

Recall – Inference For General Graphs

- Junction tree algorithm is an exact inference method for arbitrary graphs
  - A particular tree structure defined over cliques of variables
  - Inference ends up being exponential in maximum clique size
  - Therefore slow in many cases

- Sampling methods: represent desired distribution with a set of samples, as more samples are used, obtain more accurate representation

Outline

- Sampling
- Rejection Sampling
- Importance Sampling
- Markov Chain Monte Carlo

Sampling

- The fundamental problem we address in this lecture is how to obtain samples from a probability distribution \( p(z) \)
  - This could be a conditional distribution \( p(z|e) \)

- We often wish to evaluate expectations such as

\[
\mathbb{E}[f] = \int f(z) p(z) dz
\]

  - e.g. mean when \( f(z) = z \)

- For complicated \( p(z) \), this is difficult to do exactly, approximate as

\[
\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)})
\]

  where \( \{z^{(l)} | l = 1, \ldots, L\} \) are independent samples from \( p(z) \)
**Sampling**

- Approximate

\[ f = \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)}) \]

where \{z^{(l)} | l = 1, \ldots, L\} are independent samples from \( p(z) \)

---

**Bayesian Networks - Generating Fair Samples**

- How can we generate a *fair* set of samples from this BN?

---

**Sampling from Bayesian Networks**

- Sampling from discrete Bayesian networks with no observations is straight-forward, using *ancestral sampling*

Bayesian network specifies factorization of joint distribution

\[ P(z_1, \ldots, z_n) = \prod_{i=1}^{n} P(z_i | pa(z_i)) \]

- Sample in-order, sample parents before children
  - Possible because graph is a DAG
  - Choose value for \( z_i \) from \( p(z_i | pa(z_i)) \)
Sampling From Empty Network – Example

Cloudy
Rain
Sprinkler
Wet Grass

P(C)
T
F
.50
.30

P(R|C)
C
T
F
.80
.20

P(S|C)
S
R
T
T
F
T
F
F
T
F
.90
.90
.99

P(W|S,R)

P(C)
.50
.01

From Russell and Norvig, AIMA
This sampling procedure is fair, the fraction of samples with a particular value tends towards the joint probability of that value.

Define $S_{PS}(z_1,\ldots,z_n)$ to be the probability of generating the event $(z_1,\ldots,z_n)$

- This is equal to $p(z_1,\ldots,z_n)$ due to the semantics of the Bayesian network.

Define $N_{PS}(z_1,\ldots,z_n)$ to be the number of times we generate the event $(z_1,\ldots,z_n)$

$$\lim_{N\to\infty} \frac{N_{PS}(z_1,\ldots,z_n)}{N} = S_{PS}(z_1,\ldots,z_n) = p(z_1,\ldots,z_n)$$

Note that this procedure can be applied to generate samples for marginals as well.

- Simply discard portions of sample which are not needed.
- e.g. For marginal $p(\text{rain})$, sample $(\text{cloudy} = t, \text{sprinkler} = f, \text{rain} = t, \text{wg} = t)$ just becomes $(\text{rain} = t)$
- Still a fair sampling procedure.
Sampling with Evidence

- What if we observe some values and want samples from \( p(z|e) \)?
- Naive method, logic sampling:
  - Generate \( N \) samples from \( p(z) \) using ancestral sampling
  - Discard those samples that do not have correct evidence values
  - e.g. For \( p(\text{rain}|\text{cloudy} = t, \text{spr} = t, \text{wg} = t) \), sample \( (\text{cloudy} = t, \text{spr} = f, \text{rain} = t, \text{wg} = t) \) discarded
- Generates fair samples, but wastes time
  - Many samples will be discarded for low \( p(e) \)

Rejection Sampling

- Consider the case of an arbitrary, continuous \( p(z) \)
- How can we draw samples from it?
- Assume we can evaluate \( p(z) \), up to some normalization constant
  \[
  p(z) = \frac{1}{Z_p} \tilde{p}(z)
  \]
  where \( \tilde{p}(z) \) can be efficiently evaluated (e.g. MRF)

Proposal Distribution

- Let's also assume that we have some simpler distribution \( q(z) \) called a proposal distribution from which we can easily draw samples
  - e.g. \( q(z) \) is a Gaussian
- We can then draw samples from \( q(z) \) and use these
- But these wouldn't be fair samples from \( p(z) \)!

Other Problems

- Continuous variables?
  - Gaussian okay, Box-Muller and other methods
  - More complex distributions?
- Undirected graphs (MRFs)?
Comparison Function and Rejection

- Introduce constant $k$ such that $kq(z) \geq \tilde{p}(z)$ for all $z$
- Rejection sampling procedure:
  - Generate $z_0$ from $q(z)$
  - Generate $u_0$ from $[0, kq(z_0)]$ uniformly
  - If $u_0 > \tilde{p}(z)$ reject sample $z_0$, otherwise keep it
- Original samples are uniform in grey region
- Kept samples uniform in white region – hence samples from $p(z)$

Rejection Sampling Analysis

- How likely are we to keep samples?
- Probability a sample is accepted is:
  \[
  p(\text{accept}) = \int \frac{\tilde{p}(z)}{kq(z)} q(z) dz = \frac{1}{k} \int \tilde{p}(z) dz
  \]
- Smaller $k$ is better subject to $kq(z) \geq \tilde{p}(z)$ for all $z$
  - If $q(z)$ is similar to $\tilde{p}(z)$, this is easier
  - In high-dim spaces, acceptance ratio falls off exponentially
  - Finding a suitable $k$ challenging

Discretization

- Importance sampling is a sampling technique for computing expectations:
  \[
  \mathbb{E}[f] = \int f(z)p(z)dz
  \]
- Could approximate using discretization over a uniform grid:
  \[
  \mathbb{E}[f] \approx \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)})p(z^{(l)})
  \]
  - c.f. Riemannian sum
  - Much wasted computation, exponential scaling in dimension
  - Instead, again use a proposal distribution instead of a uniform grid

Importance sampling

- Approximate expectation
  \[
  \mathbb{E}[f] = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz 
  \approx \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)})\frac{p(z^{(l)})}{q(z^{(l)})}
  \]
  - Quantities $p(z^{(l)})/q(z^{(l)})$ are known as importance weights
  - Correct for use of wrong distribution $q(z)$ in sampling
Likelihood Weighted Sampling

- Consider the case where we have evidence $e$ and again desire an expectation over $p(x|e)$.
- If we have a Bayesian network, we can use a particular type of importance sampling called likelihood weighted sampling:
  - Perform ancestral sampling
  - If a variable $z_i$ is in the evidence set, set its value rather than sampling
- Importance weights are:
  \[ \frac{p(z^{(i)})}{q(z^{(i)})} = ? \]
  \[ \frac{p(z^{(i)})}{q(z^{(i)})} = \frac{p(x,e)}{p(e)} \prod_{i \in e} \frac{1}{p(z_i|pa_i)} \propto \prod_{i \in e} p(z_i|pa_i) \]

Likelihood Weighted Sampling Example

\[ w = 1.0 \]

From Russell and Norvig, AIMA
Sampling Rejection Sampling Importance Sampling Markov Chain Monte Carlo

Likelihood Weighted Sampling Example

Cloudy
Rain
Sprinkler
Wet
Grass
C
T
F
.80
.20
P(R|C)C
T
F
.10
.50
P(S|C)
S
R
T
T
T
F
F
T
F
F
F
.90
.90
.99
P(W|S,R)

P(C)
.50
.01

\[ w = 1.0 \times 0.1 \]
from Russell and Norvig, AIMA

Note that importance sampling, e.g. likelihood weighted sampling, gives approximation to expectation, not samples
But samples can be obtained using these ideas
Sampling-importance-resampling uses a proposal distribution \( q(z) \) to generate samples
Unlike rejection sampling, no parameter \( k \) is needed

Sampling Importe Resampling

SIR - Algorithm

• Sampling-importance-resampling algorithm has two stages
• Sampling:
  • Draw samples \( z^{(1)}, \ldots, z^{(L)} \) from proposal distribution \( q(z) \)
• Importance resampling:
  • Put weights on samples
    \[ w_l = \frac{p(z^{(l)})/q(z^{(l)})}{\sum_m p(z^{(m)})/q(z^{(m)})} \]
  • Draw samples from the discrete set \( z^{(1)}, \ldots, z^{(L)} \) according to weights \( w_l \) (uniform distribution)
• This two stage process is correct in the limit as \( L \to \infty \)
Markov Chain Monte Carlo

- Markov chain Monte Carlo (MCMC) methods also use a proposal distribution to generate samples from another distribution.
- Unlike the previous methods, we keep track of the samples generated $z^{(1)}, \ldots, z^{(\tau)}$.
- The proposal distribution depends on the current state: $q(z|z^{(\tau)})$.
  - Intuitively, walking around in state space, each step depends only on the current state.

Metropolis Algorithm

- Simple algorithm for walking around in state space:
  - Draw sample $z^* \sim q(z|z^{(\tau)})$.
  - Accept sample with probability
    $$A(z^*, z^{(\tau)}) = \min \left(1, \frac{p(z^*)}{p(z^{(\tau)})} \right).$$
  - If accepted $z^{(\tau+1)} = z^*$, else $z^{(\tau+1)} = z^{(\tau)}$.
- Note that if $z^*$ is better than $z^{(\tau)}$, it is always accepted.
- Every iteration produces a sample.
  - Though sometimes it's the same as previous.
  - Contrast with rejection sampling.
- The basic Metropolis algorithm assumes the proposal distribution is symmetric $q(z_A|z_B) = q(z_B|z_A)$.

Metropolis Example

- $p(z)$ is anisotropic Gaussian, proposal distribution $q(z)$ is isotropic Gaussian.
  - Red lines show rejected moves, green lines show accepted moves.
- As $\tau \to \infty$, distribution of $z^{(\tau)}$ tends to $p(z)$.
  - True if $q(z_A|z_B) > 0$.
  - In practice, burn-in the chain, collect samples after some iterations.
  - Only keep every $M^{th}$ sample.

Metropolis Example - Graphical Model

- Consider running Metropolis algorithm to draw samples from $p(\text{cloudy}, \text{rain}|\text{spr} = t, \text{wg} = t)$.
- Define $q(z|z^*)$ to be uniformly pick from $\text{cloudy, rain}$, uniformly reset its value.
**Metropolis Example**

- Walk around in this state space, keep track of how many times each state occurs.

**Gibbs Sampling**

- A simple coordinate-wise MCMC method.
- Given distribution $p(z) = p(z_1, \ldots, z_M)$, sample each variable (either in pre-defined or random order):
  - Sample $z_1^{(r+1)} \sim p(z_1 | z_2^{(r)}, \ldots, z_M^{(r)})$
  - Sample $z_2^{(r+1)} \sim p(z_2 | z_1^{(r+1)}, z_3^{(r)}, \ldots, z_M^{(r)})$
  - ...
  - Sample $z_M^{(r+1)} \sim p(z_M | z_1^{(r+1)}, z_2^{(r+1)}, \ldots, z_{M-1}^{(r+1)})$
- These are easy if Markov blanket is small, e.g. in MRF with small cliques, and forms amenable to sampling.

**Metropolis-Hastings Algorithm**

- A generalization of the previous algorithm for asymmetric proposal distributions known as the Metropolis-Hastings algorithm.
- Accept a step with probability:
  $$A(z^*, z^{(r)}) = \min \left(1, \frac{p(z^*)q(z^{(r)}|z^*)}{p(z^{(r)})q(z^*|z^{(r)})} \right)$$
- A sufficient condition for this algorithm to produce the correct distribution is detailed balance.
• Consider running Gibbs sampling on
\[ p(\text{cloudy}, \text{rain}|\text{spr} = t, \text{wg} = t) \]
• \[ q(z|z^*) \]: pick from cloudy, rain, reset its value according to
\[ p(\text{cloudy}|\text{rain, spr, wg}) \text{ (or } p(\text{rain|cloudy, spr, wg}) \]
• This is often easy – only need to look at Markov blanket

Conclusion

• Readings: Ch. 11.1-11.3 (we skipped much of it)
• Sampling methods use proposal distributions to obtain samples from complicated distributions
• Different methods, different methods of correcting for proposal distribution not matching desired distribution
• In practice, effectiveness relies on having good proposal distribution, which matches desired distribution well