CS 6347

Lecture 10

Sampling Methods
Sampling vs. Variational Methods

• Sampling:
  – Guaranteed to approach the correct answer in the limit
  – Can be quite slow to converge

• Variational methods:
  – Only approximate the true solution
  – Possible to make them quite fast
Sampling: The Basics

\[ p(x_1, ..., x_n) = \frac{1}{Z} \prod_c \psi_c(x_c) \]

• Idea: if we could generate i.i.d. samples from \( p \), we could use them to estimate the partition function, marginals, etc.

• A **sample** is an instantiation/assignment of a value for each of the random variables

\[ x^t = (x_1^t, ..., x_n^t) \]
Sampling: The Basics

\[
p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_c \psi_c(x_c)
\]

- Given \( T \) i.i.d. samples \( x^1, \ldots, x^T \) drawn from the distribution \( p \), we could estimate marginal probabilities.

- But how do we generate samples from a distribution?
Sampling: The Basics

• Let's begin with a simple example

  – Suppose we want to sample from a univariate probability distribution, \( q(y) \), where \( y \in \{1, \ldots, k\} \)

  – Sampling algorithm:

    • Divide the unit interval into \( k \) pieces corresponding to the probabilities \( q(1), \ldots, q(k) \)

    \[
    \begin{array}{c|c|c|c|c|c|c|c|c}
    & q(1) & q(2) & \ldots & q(k-1) & q(k) \\
    \hline
    \end{array}
    \]

    • Pick a random number \( z \) in \([0,1]\)

    • If \( z \) is in the \( j^{th} \) box, return \( j \)
Sampling: Bayesian Networks

• We can use the same idea to sample from (discrete) Bayesian networks

  – Sample the variables one at a time, in topological order

  – Because of the graph structure, we only have to sample from univariate (conditional) distributions!
Sampling: Bayesian Networks

A | \( P(A) \)
---|---
0 | 0.3
1 | 0.7

B | \( P(B) \)
---|---
0 | 0.4
1 | 0.6

\[
P(A|B) = \begin{cases} 
0.3 & \text{if } B = 0 \\
0.7 & \text{if } B = 1 
\end{cases}
\]

\[
P(B|A) = \begin{cases} 
0.4 & \text{if } A = 0 \\
0.6 & \text{if } A = 1 
\end{cases}
\]

\[
P(C|A, B) = \begin{cases} 
0.1 & \text{if } A = 0, B = 0 \\
0.9 & \text{if } A = 0, B = 1 \\
0.2 & \text{if } A = 1, B = 0 \\
0.8 & \text{if } A = 1, B = 1 
\end{cases}
\]

\[
P(D|C) = \begin{cases} 
0.3 & \text{if } C = 0 \\
0.7 & \text{if } C = 1 
\end{cases}
\]

random numbers: 0.8663, 0.0253, 0.1714, 0.8309
Monte Carlo Methods

- Express the estimation problem as the expectation of a random variable

\[ E_p[f(x)] = \sum_x f(x) \cdot p(x) \]

- To estimate this expectation, draw samples \(x^1, \ldots, x^T\) i.i.d. from \(p\) and approximate the expectation as

\[ \hat{f} = \sum_t \frac{f(x^t)}{T} \]
Monte Carlo Methods

- **Law of Large Numbers:** as $T \to \infty$,

\[
\sum_{t} \frac{f(x^t)}{T} \to E_p[f(x)]
\]

- $\hat{f}$ is an **unbiased estimator** of $E_p[f(x)]$

- $\text{var}(\hat{f}) = \text{var} \left( \sum_t \frac{f(x^t)}{T} \right) = \frac{\text{var}(f(x))}{T}$
  - More samples means less variance
• Suppose that we have a joint distribution $p(x, y)$ and we would like to estimate $p(y)$

  – Express this as an expectation

  \[ p(y) = \sum_{x', y'} \delta(y' = y) \cdot p(x', y') \]

  – We can then use the previous sampling strategy to estimate this expectation
Rejection Sampling

- **Rejection sampling:**
  - To estimate $p(y)$, first draw samples from $p(x', y')$ and discard those for which $y^t \neq y$
  - This process can fail miserably if $p(y)$ is very small

- Let $z^t$ be a random variable that indicates whether or not the $t^{th}$ sample from $p(x', y')$ was accepted

- $E[\sum_{t=1}^{T} z^t] = T \cdot p(y)$
Importance Sampling

- Introduce a proposal distribution \( q(x) \) such that \( p(x, y) > 0 \) implies that \( q(x) > 0 \)

\[
p(y) = \sum_x p(x, y)
\]
\[
= \sum_x p(x, y) \frac{q(x)}{q(x)}
\]
\[
= \sum_x \frac{p(x, y)}{q(x)} q(x)
\]
\[
= E_q \left[ \frac{p(x, y)}{q(x)} \right]
\]
Importance Sampling

- Draw samples from $q(x)$
  
  - Note that we can never generate a sample that occurs with probability zero
  
  - Use the samples from $q$ to approximate $p(y)$

$$p(y) \approx \frac{1}{T} \sum_{t} \frac{p(x^t, y)}{q(x^t)}$$
Sampling: Bayesian Networks

<table>
<thead>
<tr>
<th>A</th>
<th>P(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.3</td>
</tr>
<tr>
<td>1</td>
<td>.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.4</td>
</tr>
<tr>
<td>1</td>
<td>.6</td>
</tr>
</tbody>
</table>

| A | B | C | P(C|A, B) |
|---|---|---|---------|
| 0 | 0 | 0 | .1      |
| 0 | 0 | 1 | .9      |
| 0 | 1 | 0 | .2      |
| 0 | 1 | 1 | .8      |
| 1 | 0 | 0 | 0       |
| 1 | 0 | 1 | 1       |
| 1 | 1 | 0 | .25     |
| 1 | 1 | 1 | .75     |

| C | D | P(D|C) |
|---|---|-------|
| 0 | 0 | .3    |
| 0 | 1 | .7    |
| 1 | 0 | .4    |
| 1 | 1 | .6    |

Estimate $p(D = 1)$ using $q(A, B, C)$ uniform over $A, B, C$
Importance Sampling

• The proposal distribution should be close as possible to $p(x|y)$
  
  — Often, this requires knowing an analytic form of the distribution $p$
  
  • If we had that, we wouldn't need to sample!
  
  — Picking good proposal distribution is more "art" than science
Sampling from Conditional Distributions

- Can we use the same ideas to sample from conditional distributions?

\[ p(x|y) = \frac{\sum_z p(x, y, z)}{p(y)} \]

- Using sampling to estimate the numerator and denominator can produce very bad estimates

- For example, if we overestimate the numerator and underestimate the denominator
Normalized Importance Sampling

- Rewrite the conditional distribution as

\[ p(x|y) = \frac{\sum_{x',z} \delta(x' = x)p(x',y,z)}{\sum_{x',z} p(x',y,z)} \]

- Can use the same proposal distribution to sample from the numerator and the denominator
  - Common random numbers reduce the variance
Beyond Monte Carlo Methods

• All of the methods discussed so far can have serious limitations depending on the quantity being estimated

• Idea: instead of having a single proposal distribution, why not have an adaptive proposal distribution that depends on the previous sample?

\[ q(x|x') \] where \( x' \) is the previous sample and \( x \) is the new assignment to be sampled