Importance sampling

\[ Z = \sum_x \exp \left( \sum_i w_i N_i(x) \right) \]

\[ = \sum_x \exp \left( \sum_i w_i N_i(x) \right) \frac{Q(x)}{Q(x)} \]

\[ = \mathbb{E} \left[ \frac{\exp \left( \sum_i w_i N_i(x) \right)}{Q(x)} \right] \]

- Given samples \( x^{(1)}, x^{(2)}, \ldots, x^{(N)} \) drawn from \( Q \)

\[ \approx \frac{1}{N} \sum_{j=1}^{N} \frac{\exp \left( \sum_i w_i N_i(x^{(j)}) \right)}{Q(x^{(j)})} \]
Lifted Importance sampling

\[ R(x) \lor S(y), w_1 \]
\[ \neg S(y) \lor T(z), w_2 \]

Easy to do: Replace the conditioning step by sampling
Lifted Importance Sampling: Issues

- Until now everything works fine
- How to design “Q?”
  - Apply Rules even if they cannot be applied
  - $R(x, y) \lor S(y, z) \lor T(z, u), w$
    - Use singleton rule on $R(x,y)$ even though it is not a singleton
  - This yields a tractable lifted AND/OR search tree
  - Put probabilities on the OR to AND edges. Yields a tractable distribution!
- Complexity of generating a sample
  - Linear in the height; logarithmic: max branching factor
WEBKB domain, #objects = 300

Lower Bound on log(Z)

Time in seconds

SAMPLING-BASED INFERENCE IN PRESENCE OF LOGICAL STRUCTURE
Entity Resolution domain, #objects = 300

Lower Bound on log(Z)

Time in seconds

Better
Lifted Importance Sampling: Issues

- Issue 2: How to compute the weight?
- Once you generate a sample, you need to weigh it to estimate the partition function
- \( F = R(x, y) \lor S(y, z) \lor T(x, u), w \)
  - Think of the sample as a database
- Weight of the sample
  - \( \exp(\text{number of groundings of } F \text{ that are true} \times w) \)
  - Naïve algorithm: \( O(n^4) \)
  - Database techniques: \( O(n^2) \)

\[
\approx \frac{1}{N} \sum_{j=1}^{N} \exp \left( \sum_i w_i N_i(x^{(j)}) \right) / Q(x^{(j)})
\]

<table>
<thead>
<tr>
<th>( R(x, y) )</th>
<th>( S(y, z) )</th>
<th>( T(z, u) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R(1, 1) )</td>
<td>( S(2, 1) )</td>
<td>( T(2, 1) )</td>
</tr>
<tr>
<td>( R(2, 1) )</td>
<td>( S(3, 2) )</td>
<td>( T(3, 2) )</td>
</tr>
<tr>
<td>( R(3, 1) )</td>
<td>( S(3, 4) )</td>
<td>( T(2, 2) )</td>
</tr>
<tr>
<td>( R(1, 4) )</td>
<td>( S(2, 4) )</td>
<td>( S(2, 4) )</td>
</tr>
</tbody>
</table>
Complexity of finding the number of groundings is exponential in the treewidth of the following graph:

- A vertex for each logical variable.
- If two logical variables appear in a predicate, they are connected by an edge.

\[ F = R(x, y) \lor S(y, z) \lor T(z, u), w \]

Nothing but a Markov network with three potentials.

Theorem: The complexity of lifted importance sampling is exponential in the treewidth of the largest formula!

Too large for many practical applications.
Approximate Counting

- If treewidth of a formula is too large
  - Use approximate counting methods.
    - SampleSearch (Gogate & Dechter, 2011) or the Wish Algorithm (Ermon et al. 2014) and many more.
    - Belief propagation approaches
    - Problem is structured, better approximations possible

- Many approximations yield asymptotically unbiased estimate instead of unbiased estimates
Gibbs sampling for MLNs

- Input: Ground MLN
  - Start with a random assignment to all propositions
  - Select a proposition $A$ uniformly at random
    - Compute $Pr(A \mid \text{assignment to all others})$
    - Resample $A$ from $Pr(A \mid \text{assignment to all others})$
Gibbs sampling for MLNs

\[ R(x, y) \lor S(y, z), w_1 \]
\[ S(y, z) \lor T(z, u), w_2 \]

<table>
<thead>
<tr>
<th></th>
<th>( R(x, y) )</th>
<th>( S(y, z) )</th>
<th>( T(z, u) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( R(1,1) )</td>
<td>( S(2,1) )</td>
<td>( T(2,1) )</td>
</tr>
<tr>
<td>2</td>
<td>( R(2,1) )</td>
<td>( S(3,2) )</td>
<td>( T(3,2) )</td>
</tr>
<tr>
<td>3</td>
<td>( R(3,1) )</td>
<td>( S(3,4) )</td>
<td>( T(2,2) )</td>
</tr>
<tr>
<td>4</td>
<td>( R(1,4) )</td>
<td>( S(2,4) )</td>
<td>( S(2,4) )</td>
</tr>
</tbody>
</table>

Pick \( R(1,1) \) to flip

\[ \Pr(R(1,1) = 1 | \text{all others}) \propto e^{N_1(\text{database})w_1 + N_2(\text{database})w_2} \]

\[ \Pr(R(1,1) = 0 | \text{all others}) \propto e^{N_1(\text{database}')w_1 + N_2(\text{database}')w_2} \]

Where \( \text{database}' \) is same as database except that \( R(1,1)=0 \)

- Database theory described before can be applied here.
- Efficient version of WALKSAT (Selman, Kautz & Cohen, ’93) using the same approach
LIFTING THE GIBBS SAMPLING ALGORITHM

- Lifted sample
  - Rather than storing the precise groundings of an atom that are true in the sample only store “the number of groundings that are true.”
  - Not always possible

- Gibbs sampling is not inherently liftable
  - Blocked Gibbs sampling is more amenable to lifting

- Blocking (Jensen-Kjaerulff-Kong, 1993)
  - Jointly sample a set of atoms (blocks)
  - Tradeoffs: Increasing the block size increases complexity but reduces the error

Venugopal&Gogate,’12; Niepert, ‘12
LIFTED BLOCKED GIBBS SAMPLING

- Time complexity of running exact inference = $O(n^3)$
- Space complexity = $O(n^2)$

- Higher accuracy (more atoms in a cluster) and (counter-intuitively) smaller complexity!

$R(x,y) \lor S(y,z), w_1$

$S(y,z) \lor T(z,u), w_2$
Algorithm LBG(MLN M)

- **Clustering**: Partition atoms in M into mutually exclusive and exhaustive clusters (blocks) such that lifted-search is polynomial within each cluster.

- **Cluster Graph**: Connect clusters if they have atoms that are in Markov Blanket of each other.

- **Message-passing**:
  - Each incoming message is assignment to a subset of the Markov Blanket of atoms in the cluster.
  - Message can be represented in a lifted manner.
Evidence Problem

- Serious problem with inference: “Evidence-problem”
  - \[1.75 \neg \text{Strong}(x) \lor \text{Wins}(x,y)\]

<table>
<thead>
<tr>
<th>Wins(A,A)</th>
<th>0.56</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wins(A,B)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(A,C)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(B,A)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(B,B)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(B,C)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(C,A)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(C,B)</td>
<td>0.56</td>
</tr>
<tr>
<td>Wins(C,C)</td>
<td>0.56</td>
</tr>
</tbody>
</table>

- Evidence breaks symmetries that are essential for lifting
  - Lifted algorithms ground/shatter \(\Rightarrow\) Inference complexity increases

- Reasoning with evidence is critical to almost every application

Van den Broeck & Darwiche, ’13; Venugopal & Gogate, ’14, Beltagy & Mooney, ’14
Our approach

- What is needed?
  - Scalable, practical inference that works out-of-the-box without restricting MLN/Evidence structure
  - “Compress” the MLN by clustering related objects
    - “Scales-up” existing inference algorithms
  - Control complexity using a parameter (#clusters)
    - #Clusters trades-off inference accuracy with complexity
Clustering

- Domain-reduction as a clustering problem

\[ \neg R(x, y) \lor \neg S(y, z) \lor T(z, x) \]

Domain of each logical-variable is reduced from “n” to some number much smaller than “n”

Formulate as clustering problem by developing a distance function and minimizing with respect to it.
Evidence-Specific Distance Function

- The MLN representation has an inherent symmetry
  - “Similar” evidence on distinct objects likely translates to similar marginal probabilities
- Distance function $(X_i, X_k)$
  - Sum of the difference between the number of groundings of a formula satisfied by $X_i$ and by $X_k$
  - Any distance function can be used depending on the application
Experiments

- Clustering algorithms from Weka
  - Kmeans++ (KM)
  - Hierarchical Clustering (HC)
  - Expectation Maximization (EM)
  - Xmeans

- Inference algorithms from Alchemy
  - Lifted Belief Propagation (LBP)
  - Gibbs Sampling (Gibbs)
Results on WebKB

KL-Divergence from marginals computed on the original MLN

Inverse Compression Ratio (ICR) = \#Compressed-Ground-Formulas/\#Original-Ground-Formulas
Entity Resolution

![Graphs showing error vs. ICR for different methods: KM, XM, EM, HC, LBP, and Gibbs.](image)