Inference: Miscellaneous

Daniel Lowd
May 22, 2013
Review: Exact Inference

• Brute force
  – IDEA: Multiply and add
  – Exponential in the number of variables.

• Variable elimination
  – IDEA: A smarter way to multiply and add
  – Exponential in the treewidth of the induced graph.

• Belief propagation in trees
  – IDEA: Variable elimination in all directions at once

• Junction tree algorithm (briefly covered):
  – IDEA: Run BP in a tree of cliques satisfying the “running intersection” property.
Review: Approximate Inference

• Loopy belief propagation
  – IDEA: Run belief propagation in a loopy graph and hope for the best
  – Bad when the network has tight loops with strong interactions

• Gibbs sampling
  – IDEA: Wander around randomly, resampling one variable at a time given Markov blanket. Average.
  – Bad when transitioning between modes is very unlikely.
  – Usually much better than rejection sampling and likelihood weighting.
Variational Inference

• You have a hard probability distribution $P$. What to do?
• Pick an “easy” distribution $Q$, e.g.:
  
  \[ Q(X_1, \ldots, X_n) = Q(X_1)Q(X_2)\ldots Q(X_n) \]
• Pick the parameters of $Q$ so that it is “close” to the distribution you want, $P$:
  \[
  \min_Q D(P, Q)
  \]
• Use $Q$ to answer whatever questions you want.
How to find Q

• Typically minimize “reverse” KL-divergence:

\[ KL(Q \parallel P) = \sum_x Q(x) \log \frac{P(x)}{Q(x)} \]

• Updates are simple: (relatively speaking)

\[ Q(X_i) \leftarrow \exp \left( E_{X_{-i} \sim Q} \left[ \log P(X_i \mid X_{-i}) \right] \right) / Z_i \]

• Finds local optimum, which typically represents one mode of the probability distribution.

• Gives you a lower bound on the partition function. (Lots of work focuses on this aspect!)
Maximum a Posteriori (MAP) Inference

- **Goal**: find most likely state of all variables.
- Iterated conditional modes:
  Like Gibbs sampling, but always pick most likely state until you reach a local optimum.
- Max-product algorithm:
  Like BP, but use max instead of sum.
- Graph cuts:
  If all potentials are associative, MAP solution can be found in polynomial time with min-cut algorithm.
Detour: Log-linear Models

Product of factors can be represented as an exponentiated sum of log-factors:

\[ P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_i \phi_1(D_i) \]

\[ \log P(X_1, \ldots, X_n) = -\log Z + \sum_i \log \phi_1(D_i) \]

\[ P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left( \sum_i \log \phi_1(D_i) \right) \]
Log-linear Models

Log-factors can be represented as weighted functions, sometimes much more compactly:

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( \varphi_1(A, B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>10</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>1</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>1</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>10</td>
</tr>
</tbody>
</table>

\[
\log \left( \log 10 \right) f_{A=B}(A, B) = (\log 10) f_1(A, B) + (\log 1) f_2(A, B) + (\log 1) f_3(A, B) + (\log 10) f_4(A, B)
\]
Log-linear Models

Putting it all together...

\[ \log P(X_1, \ldots, X_n) = \sum_i w_i f_i(D_i) - \log Z \]

How do we maximize this?
MAP is an optimization problem

• Use integer linear programming
• Use weighted MAX-SAT solvers
  – Encode each feature as a clause.
  – Weighted MAX-SAT tries to maximize the sum of satisfied clause weights.

• Dual-decomposition
  – Maximize each factor (or set of factors) separately
  – Adjust factor parameters to encourage them to agree.
  – Similar ideas work for computing marginals, too!
Themes in Inference

• Optimization
• Relaxations
• Bounds
• Exploiting structure