Dynamic Programming & Graphical Model

Graphical Models for prob. dist.

A way to define high dimensional prob dist.

Factor Graphs an example first

Ex. Medical diagnosis

\( C = \begin{cases} 
1 & \text{if I have a cold} \\
0 & \text{o.w.}
\end{cases} \)

\( S = \begin{cases} 
1 & \text{if I have a sore throat} \\
0 & \text{o.w.}
\end{cases} \)

\( Y = \begin{cases} 
1 & \text{if I have a runny nose} \\
0 & \text{o.w.}
\end{cases} \)

\( P = \begin{cases} 
1 & \text{if there is pollen in the air} \\
0 & \text{o.w.}
\end{cases} \)

\[
\begin{array}{ccc}
C & S & Y & \Psi_a(c, s, y) \\
1 & 1 & 1 & .1 \\
0 & 0 & 1 & .3 \\
0 & 0 & 0 & .6
\end{array}
\]

\[
\begin{array}{ccc}
Y & P & \Psi_b(y, p) \\
0 & 1 & .1 \\
1 & 0 & .2 \\
1 & 1 & .3 \\
0 & 0 & .4
\end{array}
\]
Assume $\psi_a$ & $\psi_b$ ind unless told O.w. 2

$Pr(b = 0, s = 0, r = 1, p = 1) = (.3)(.3) = .09$

$P(c, s, r, p) = \frac{1}{Z} \psi_a(c, s, r) \psi_b(r, p)$

$Z$ = normalization constant.

In our example

<table>
<thead>
<tr>
<th></th>
<th>1111</th>
<th>1110</th>
<th>0011</th>
<th>0010</th>
<th>0001</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.03</td>
<td>.02</td>
<td>.09</td>
<td>.06</td>
<td>.06</td>
<td>.04</td>
</tr>
</tbody>
</table>

$Z = .50$

View as a graph = factor graph

```
  C
 /  \
\psi_a / \\
  \  \\
S    Y
  \\
  /  \
\psi_b / \\
  \  \\
P
```
In general: Factor Graph is a bipartite

The graph will have:
1) Variable nodes: \( V \)
2) Factor nodes: \( a \) is \( da \equiv \text{neig of } a \) where \( da \equiv \text{variables of } \psi_a \).

3) For each factor a map:
\[
\psi_a : \{0,1\}^{\text{Da}} \rightarrow \mathbb{R}^+ 
\]

**Tasks**: Given a factor graph \( G \).

1) Compute marginals \( \text{ie } P(X_i = 1) \)
2) Compute conditionals \( \text{ie } P(X_i = 1 \mid X_j = 0) \)
3) Same from dist.
4) Find mode \( \text{ie } \arg\max \{ P(x) \} \)
Claim. For general factor graph computing any of the above is NP-hard.

We will show: 3-SAT ≤ₚ Marginal

the others are similar.

Recall: 3-CNF is a Boolean formula, with 3 literals per clause.

3-SAT = \{ ϕ ∈ 3-CNF | ϕ has a satisfying instance \}

Example: \( F = (x_1 \lor x_2 \lor x_3) \land (x_3 \lor x_4 \lor x_5) \) ∈ 3-CNF

Note \( (x_1 \lor \neg x_2 \lor x_3) \) is false iff \( x_1 = \bar{x}_3 = \top \) & \( x_2 = \bot \)

Suppose \( ϕ ∈ 3-CNF \) with clauses \( C_1, \ldots, C_m \).

for each clause \( C \) let \( \psi_C \) be the factor s.t.

\[ \psi_C \equiv \begin{cases} 1 & \text{if true} \\ 0 & \text{otherwise} \end{cases} \]

In our example:

\[ \psi_{C_1}(x_1, x_2, x_3) = \begin{cases} 1 & \text{if true} \\ 0 & \text{otherwise} \end{cases} \]
Consider uniform dist over \( \{0,1\}^5 \)

\[
P(x_1, \ldots, x_5) = \frac{1}{Z} \psi_{c_1}(x_1, x_2, x_3) \psi_{c_2}(x_3, x_4, x_5)
\]

When \( Z \neq 0 \) \( P \) is uniform dist over satisfying assignments.

Thus \( P(x_1 = 1) > 0 \) iff \( \Phi \) is satisfiable.

Thus computing marginals is \( \text{NP-Hard} \).

Note: Computing \( Z \) is \( \#P\)-complete.
Factor Graphs that are trees

We restrict factor graphs to a tree.

To show For trees factor graph tasks \(1, \ldots, 4\) are poly-time.

We will solve the unnormalized prob

\[
Z = \sum_{x \in \{0,1\}^n} \prod_{a=1}^{m} \psi_a(x_\partial a)
\]

\[
Z[x_i = b] = \sum_{x \in \{0,1\}^n} \prod_{a=1}^{m} \psi_a(x_\partial a) \\
x_i = b
\]

Note: \(P[x_i = b] = Z[x_i = b]/Z\)

Goal: Compute \(Z[x_i = b]\)

We root \(G\) at vertex \(x_i\)

Subproblems: Rooted subtrees of \(G\)
Let $T_w$ be subtree rooted at $w$ (var in fact) two types of subtrees.

1) Variable rooted 2) Factor rooted

$T_{x_i}$ be subtree of $T_x$ with child $v_i$

$V_w$ be all variables in $T_w$ ($T_{x_i}^q$)

$F_w$ be all factors in $T_w$ ($T_{x_i}^q$)

$\text{Child}(w) = \text{children}(w)$

---

**For Variable-rooted**

$V_i \in \mathfrak{V}$ \& $A \in \{0, 1\}$

**compute**: $V_i(\beta) = \sum_{y \in \{0, 1\}} \prod_{i \in F_i} \Psi_y(v_i, y_{da})$
For factor-rooted trees:

\[ \forall i \in \mathcal{V} \text{ and } \forall b \in \{0, 1\} \text{ and } \forall \bar{a} \in \text{child}(x_i) \]

Compute:

\[ V_i(\bar{a}, b) = \sum_{y_0} \psi_{\bar{a}}(y_0) \prod_{y \in \{0, 1\} \setminus y_0} \psi_{y} \prod_{a \in F_{\bar{a}}} \psi_{a}(y_{\bar{a}}) \]

\[ y_i = b \]

3) Recurrences:

For variable-rooted subtrees

\[ V_i(b) = \begin{cases} \prod_{\xi \in \text{ch}(i)} V_i(\bar{a}, b) & x_i \text{ not leaf} \\ 0 & \text{otherwise} \end{cases} \]
For factor-rooted subtrees

\[ V_i(a,b) = \begin{cases} 
\sum_{y \in \{0,1\}} \Psi_i(y_{\bar{a}}) \prod_{j \neq i} V_j(y_j) & \text{if not leaf} \\
\psi_{a}(b) & \text{if leaf} 
\end{cases} \]

4) Correction: Simple induction

Called the "Sum-Product Alg"