instance, \(x'\). In general, the \(i\)th converted factor \(\phi_i\) is not a function of the first \(i-1\) variables, since they are fixed to values in \(x'\).

By simplifying the factors, we obtain the following:

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(\phi_1(X_1, X_2))</th>
<th>(X_2)</th>
<th>(\phi_2(X_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>1</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>1/4</td>
<td>F</td>
<td>1/2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>3/2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Multiplying the factors together and renormalizing yields:

\[
P(X_1, X_2) = \frac{1}{Z(X_{-i})} \exp \left( \sum_j w_j f_j(D_j) \right)
\]

Note that the normalization \(Z\) is now a function of the evidence variables, \(X_{-i}\). To represent \(P(X_i|x^{(i)}_{-o[i]})\), we can condition each \(f_j\) on \(x^{(i)}_{-o[i]}\) separately:

\[
P(X_i|x^{(i)}_{-o[i]}) = \frac{1}{Z(x^{(i)}_{-o[i]})} \exp \left( \sum_j w_j f_j(X_i, x^{(i)}_{-o[i]}) \right)
\]

\(x^{(i)}_{-o[i]}\) uses values from \(x'\) for the first \(i\) variables in \(o\) and \(x\) for the rest. The values from \(x'\) are constant but those in \(x\) are free variables, so that the resulting distribution is a function of \(x\). When simplifying \(f_j(x^{(i)}_{-o[i]}),\) if the constant values in \(x^{(i)}_{-o[i]}\) violate one of the variable tests in \(f_j\), then \(f_j\) is always zero and it can be removed entirely. Otherwise, any conditions satisfied by constant values in \(x^{(i)}_{-o[i]}\) are always satisfied and can be removed.

### 4.2 Log-Linear Models with Conjoint Features

In the previous example, we showed how to convert a DN to an MN using tables as factors. However, this can be very inefficient when the conditional distributions have structure, such as decision trees or logistic regression models. Rather than treating each type of CPD separately, we discuss how to convert any distribution that can be represented as a log-linear model with conjunctive features.

Suppose the CPD for \(X_i\) is a log-linear model:

\[
P(X_i|X_{-i}) = \frac{1}{Z(X_{-i})} \exp \left( \sum_j w_j f_j(D_j) \right)
\]

To compute \(\phi_2\), we must additionally condition the function in the denominator on \(X_2 = x'^T_2 = T\). If the weights of \(f_1\) and \(f_2\) are \(w_1\) and \(w_2\), respectively, then:

\[
\phi_2(X_1, X_2) = \frac{P(X_2|X_1, X_2 = T)}{P(X_2 = T|X_1, X_4 = T)}
\]

\[
= \frac{\exp(w_1(X_1 \land X_2) + w_2(X_1 \land X_2)) / Z(x^{(2)}_{-o[2]})}{\exp(w_2(X_1 \land X_2)) / Z(x^{(2)}_{-o[2]})}
\]

\[
= \exp(w_1(X_1 \land X_2) + w_2(X_1 \land X_2) - w_2X_1)
\]

Note that the final factor \(\phi_i\) is always a log-linear model where each feature is a subset of one of the features in the original conditional distribution. Therefore, converting each conditional distribution in this manner yields an MN represented as a log-linear model.

We summarize our complete method for consistent DNs with the algorithms in Tables 1 and 2. DN2MN takes a set of conditional probability distributions, \(\{P_i(X_i|X_{-i})\}\), a base instance \(x'\), and an inverse variable ordering \(o^{-1}\). The inverse variable ordering is a mapping from variables to their corresponding indices in the desired ordering \(o\), so that \(o^{-1}[o[i]] = i\). DN2MN first converts the conditional

<table>
<thead>
<tr>
<th>Table 1: The Basic DN2MN Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>function DN2MN((P_i(X_i</td>
</tr>
<tr>
<td>(M \leftarrow \emptyset)</td>
</tr>
<tr>
<td>for (i = 1) to (n) do</td>
</tr>
<tr>
<td>Convert (P_i) to a set of weighted features, (F_i)</td>
</tr>
<tr>
<td>for each weighted feature ((w, f) \in F_i) do</td>
</tr>
<tr>
<td>(f_n \leftarrow \text{SIMPLIFYFEATURE}(i, f, x', o^{-1}, \text{true}))</td>
</tr>
<tr>
<td>(f_d \leftarrow \text{SIMPLIFYFEATURE}(i, f, x', o^{-1}, \text{false}))</td>
</tr>
<tr>
<td>(M = M \cup (w, f_n) \cup (-w, f_d))</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>return (M)</td>
</tr>
</tbody>
</table>

For example, suppose \(P(X_2|X_1, X_4)\) uses the following three conjunctive features:

\[
f_1(X_1, X_2, X_4) = X_1 \land \neg X_2 \land X_4
\]

\[
f_2(X_1, X_2) = X_1 \land X_2
\]

\[
f_3(X_2, X_4) = X_2 \land \neg X_4
\]

If \(x' = [T, T, T, T]\) and \(o = [1, 2, 3, 4]\), then \(x^{(2)} = [X_1, X_2, T, T]\). After conditioning \(f_1, f_2,\) and \(f_3\) on \(x^{(2)}_{-o[2]}\), they simplify to:

\[
f_1(X_1, X_2) = X_1 \land \neg X_2
\]

\[
f_2(X_1, X_2) = X_1 \land X_2
\]

\(f_3\) is removed entirely, since it is inconsistent with \(x^{(2)}_{-o[2]}\).