2.3.3　エビデンスの挿入　(エビデンスを証明とか訳すよりかエビデンスのままで)

あなたが何か新しい情報を得たときに、ベイジアンネットワークをつかって何か新しい確率を計算するとしよう。その新しい情報を得る以前の情報をタイプ A=a とする。ここで A は変数を、そして a は A の状態を表す。A は n 個の状態をもっており、それらの確率が P(A)=(x1,..,xn)と表されるとしよう。そして、A が状態 i かまたは j にあるときにのみ、情報 e が得られるものと仮定しよう。これをステートメントすると、状態 i と j を除いた他の状態は全て不可能なので、確率密度 P(A,e) = (0,…,0,xi,0,…,0,xj,0,…0)というように表されるだろう。

ここで、P(e)は e の事前確率を表し、P(A,e)を A で周辺化することで得られる。また、P(A,e)は P(A)を (0,…,0,1,0,…,0,1,0,…0)で乗じた結果である。この () 内の 1 の位置は i 番目と j 番目にあたる。

以下、スライドからの例。

Consider a variable A with five states a_1, a_2, a_3, a_4, a_5 and with probability:

$$P(A) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, \quad \sum_{i=1}^{5} x_i = 1$$

Assume that we get the evidence e: "A is either in state a_2 or a_4". Then:

$$P(A, e) = \begin{pmatrix} 0 \\ x_2 \\ 0 \\ x_4 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Thus, e can be represented by a potential $\bar{e} = (0, 1, 0, 1, 0)^T$ and:

$$P(A, e) = P(A) \cdot \bar{e}$$

定義 2.4　A を n 個の状態をもつ変数とする。このとき A の finding とは、0 と 1 からなる n 次元の表である。

ステートメント e、すなわち「A は状態 i か j のいずれかである」ということと、それに対応する０／１の finding ベクトルとの間の違いを示すために、我々はときどきそ

の finding に対して太字の **e** を用いることがある。意味的にいうと、finding とは、A のある状態が不可能であるというステートメントである。

ここで、ある結合確率テーブル P(U)があり、**e** が finding であるとしよう。結合確率テーブル P(U,e)は、P(U)のなかの A のエントリーのうち状態 i か j でない全てのエントリーを 0 でおきかえ、それ以外のエントリーはそのままにしておくことで得られるテーブルである。これは、P(U)を **e** で乗じたことと同じことである。すなわち、

$$P(U, e) = P(U) \cdot \mathbf{e}$$

となる。$P(e) = \Sigma_U P(U, e) = \Sigma_U (P(U) \cdot \mathbf{e})$ であることに留意されたい。ベイジアンネットワークの連鎖法則を用いると、次の定理を得る。

<u>Theorem:</u> Let BN be a Bayesian network over the universe $\mathcal{U} = \{A_1, \ldots, A_n\}$, and let $\bar{e}_1$, $\bar{e}_2, \ldots, \bar{e}_m$ be findings. Then:

$$\begin{aligned} P(\mathcal{U}, e) &= P(\mathcal{U}) \cdot \prod_{i=1}^{m} \bar{e}_i \\ &= \prod_{i=1}^{n} P(A_i | \mathrm{Pa}(A_i)) \prod_{j=1}^{m} \bar{e}_j. \end{aligned}$$

Hence, to find $P(A|e)$ we use:

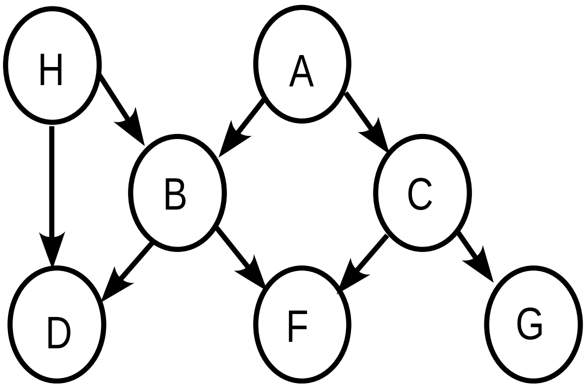
$$P(A|e) = \frac{\sum_{\mathcal{U} \setminus \{A\}} P(\mathcal{U}, e)}{P(e)}.$$

エビデンスのタイプによっては、finding として表すことはできないものもある。たとえば、あなたがだれかから、A の状態が a1 にある確率が a2 にある確率の 2 倍高いというステートメントを受け取ったとしよう。ベイジアンネットワークではこの手のエビデンスを扱うことが可能である。このとき、上述のステートメントは(0.67,0.33)の分布で表すことができ、定理 2.2 が適用できる。ただし、尤度のステートメントが真であるということが何を意味するか不明なので、P(e)をエビデンスの確率として解釈することはできない。したがって、P(U,e)のセマンティックスは不明である。我々は尤度のエビデンスについてこれ以上触れないこととする。

2.3.4　実際の確率計算

2.3.3 節で述べたように、また例 2.6 で説明したように、ベイジアンネットワークで確率を更新するということは、つまり連鎖法則をつかってユニバースの結合確率テーブル P(U)を計算するということである。ただし、U は、P(U)が手に負えないぐらい大きくなる以前に大きくある必要はない。この節で我々は、結合確率テーブルが完全にわからなくても計算が実行できることを説明する。第 4 章で、確率の更新に対するアルゴリズムを詳細に扱う予定である。

図 2.17 に示すベイジアンネットを考える。全部で 10 個の状態をもつ。ここで、我々はエビデンス e={D=d, F=f} というエビデンスをもったうえで、P(A|e)を計算してみよう。



連鎖法則から次の式を得る。

$$
\begin{aligned}
P(U,e) &= P(A,B,C,d,f,G,H) \\
&= P(A)P(H)P(B|A,H)P(C|A)P(d|B,H)P(f|B,C)P(G|C)
\end{aligned}
$$

ここで、P(d|B,H)とは、D のエントリーを状態 d に修正して得られる B と H にわたるテーブルを指す。つまり、条件付き確率テーブルは、D=d に初期化されたといいかえてもよい。ここで、我々は 10 の 7 乗ものエントリーをもつ P(U)の完全なテーブルを計算する必要はないことに注目する。もしエビデンスが入力されるまで待つとしたら、このケースでは 10 の 5 乗のエントリーしかないテーブルで作業する必要があるだろう。あとで、我々は多くても 1000 個のエントリーしかないテーブルで作業すればよいことがわかる。

P(A, e)を計算するのに、我々は P(A,B,C,d,f,G,H)から変数 B,C,G,H を周辺化する。周辺化する順番は結果に影響を与えないので、G から始めよう。

$$\Sigma_G P(A,B,C,d,f,G,H)$$
$$=\Sigma_G P(A)P(H)P(B|A,H)P(C|A)P(d|B,H)P(f|B,C)P(G|C)$$

右辺において、最後のテーブルだけが G をその領域にもっているので、分布法則（1.4 節参照）から次のようになる。

$$\Sigma_G P(A,B,C,d,f,G,H)$$
$$=P(A)P(H)P(B|A,H)P(C|A)P(d|B,H)P(f|B,C)\ \ \Sigma_G P(G|C)$$

我々は $\Sigma_G P(G|C)$ を計算する必要はない。実際、C の各状態 c に対して、我々は $\Sigma_G P(G|c)=1$ である。したがって、計算は必要なく、次の式をえる。

$$\Sigma_G P(A,B,C,d,f,G,H)$$
$$=P(A)P(H)P(B|A,H)P(C|A)P(d|B,H)P(f|B,C)$$

次に、H について周辺化する。同様にして次の式を得る。

$$\Sigma_H P(A,B,C,d,f,H)$$
$$=P(A)P(C|A)P(f|B,C)\ \ \Sigma_H P(H)P(B|A,H)P(d|B,H)$$

上の式の右辺で $P(H)$ と $P(B|A,H)$ と $P(d|B,H)$ を乗じて、それを H について周辺化する。その結果を $T(d,B,A)$ とすると、次の式をえる。

$$P(A,B,C,d,f)\ \ =P(A)P(C|A)P(f|B,C)\ \ T(d,B,A)$$

最後に、この積を計算して、それから B と C について周辺化する。　なお、5 つの変数をもつ $P(A,B,C,d,f,G,H)$ のなかで、三つより多い変数のテーブル（上の $P(H)$ と $P(B|A,H)$ と $P(d|B,H)$ の積は A,B,H の三つの変数のテーブルだから扱える）を扱えないことに留意する。我々がここで用いた手法は、変数削減と呼ばれ、次のように説明できる。

　テーブルの集合 T があるとする。これを変数 X について周辺化しよう。我々は T からその領域に X をもつ全てのテーブルを抽出し、それらの積を計算する。そして、その積を X について周辺化し、その結果のテーブルを T に加える。

＊＊スライドの方にもっとわかりやすい変数削減の例がのっている。