Making Decisions In Context

Robert M. Haralick

Computer Science, Graduate Center
City University of New York

Outline

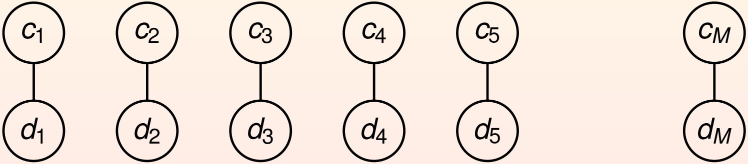
Conditional Independence Graph

Definition

A conditional independence graph $G = (V, E)$ is an undirected graph whose set V of nodes are the set of the random variables and in which no edge between node i and j, $\{i, j\} \notin E$ means variable X_i is conditionally independent of variable X_j given all the remaining variables.

No Context

$$P(c_1, d_1, \ldots, c_N, d_N) = \prod_{n=1}^{N} P(c_n, d_n)$$
$$= \prod_{n=1}^{N} P(d_n \mid c_n) P(c_n)$$

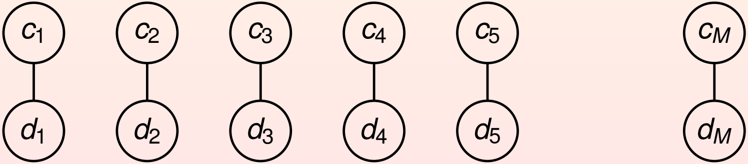


No Context

$$
\begin{aligned}
P(c_1,\ldots,c_M,d_1,\ldots,d_M) &= P(d_1,\ldots,d_M \mid c_1,\ldots,c_M)P(c_1,\ldots,c_M) \\
&= \left[\prod_{m=1}^{M} P(d_m \mid c_1,\ldots,c_M)\right]\left[\prod_{m=1}^{M} P_m(c_m)\right] \\
&= \left[\prod_{m=1}^{M} P(d_m \mid c_m)\right]\left[\prod_{m=1}^{M} P(c_m)\right] \\
&= \prod_{m=1}^{M} P(d_m \mid c_m)P(c_m)
\end{aligned}
$$

Gain Matrix: Gain 1 for a correct decision 0 for an incorrect decision.
Therefore if there are Q decisions which are correct the gain is Q.

Assign measurement d_m to class c_m where class c_m maximizes $P(d_m \mid c_m)P(c_m)$



Context

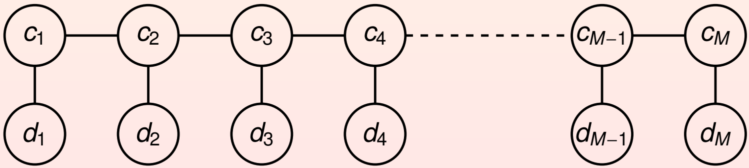
The decision making problem in context involves repeatedly making measurements of units in a situation where there is a dependency between the true classes from one unit to the next.

Definition

A decision rule using context is called a compound decision rule.

Context: Hidden Markov Model

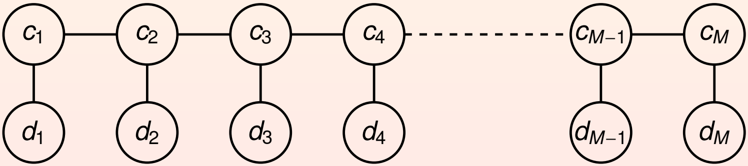
$$
\begin{aligned}
P(c_1,\ldots,c_M,d_1,\ldots,d_M) &= P(d_1,\ldots,d_M \mid c_1,\ldots,c_M)P(c_1,\ldots,c_M) \\
&= \left[\prod_{m=1}^{M} P(d_m \mid c_1,\ldots,c_M)\right] P(c_1,\ldots,c_M) \\
&= \left[\prod_{m=1}^{M} P(d_m \mid c_m)\right] P(c_1,\ldots,c_M) \\
P(c_1,\ldots,c_M) &= P(c_1 \mid c_2,\ldots,c_M)P(c_2,\ldots,c_M) \\
&= P_{1,2}(c_1 \mid c_2)P(c_2 \mid c_3,\ldots,c_M)P(c_3,\ldots,c_M) \\
&= \left[\prod_{m=1}^{M-1} P_{m,m+1}(c_m \mid c_{m+1})\right] P(c_M)
\end{aligned}
$$



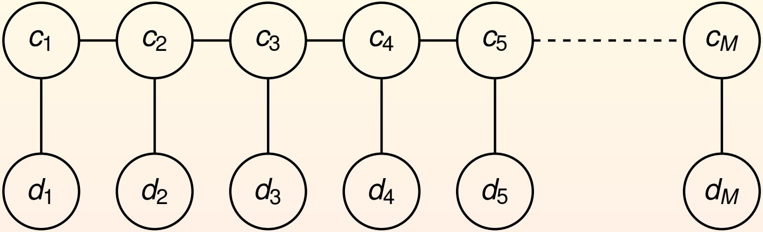
Context: Hidden Markov Model

$$
\begin{aligned}
P(c_1,\ldots,c_M,d_1,\ldots,d_M) &= \left[\prod_{m=1}^{M} P(d_m \mid c_m)\right]\left[\prod_{m=1}^{M-1} P_{m,m+1}(c_m \mid c_{m+1})\right] P(c_M) \\
&= \left[\prod_{m=1}^{M-1} P(d_m \mid c_m) P_{m,m+1}(c_m \mid c_{m+1})\right] P(d_M \mid c_M) P(c_M) \\
&= \prod_{m=1}^{M} P(d_m \mid c_m) P_{m,m+1}(c_m \mid c_{m+1})
\end{aligned}
$$

where $P_{M,M+1}(c_M \mid c_{M+1}) = P(c_M)$



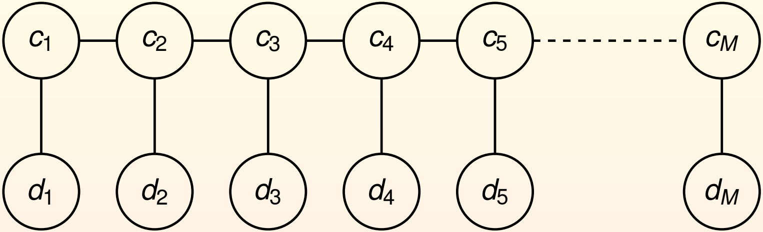
Conditional Independence Graph



Cliques: $\{d_1, c_1\}, \{c_1, c_2\}, \{c_2, d_2\}, \{c_2, c_3\}, \{c_3, d_3\} \ldots \{c_{M-1}, c_M\}, \{c_M, d_M\}$

Separators $c_1, \quad c_2, \quad c_2, \quad c_3 \quad \ldots \quad c_{M-1}, \quad c_M$

Conditional Independence Graph



$$
\begin{aligned}
P(c_1, \ldots, c_M, d_1, \ldots, d_M) &= \frac{\prod_{i=1}^{M} P(d_i, c_i) \prod_{j=1}^{M-1} P(c_j, c_{j+1})}{\prod_{k=1}^{M} P(c_k) \prod_{m=2}^{M-1} P(c_m)} \\
&= \left[\prod_{i=1}^{M} P(d_i \mid c_i)\right] \left[\prod_{j=2}^{M} P(c_j \mid c_{j-1})\right] P(c_1)
\end{aligned}
$$

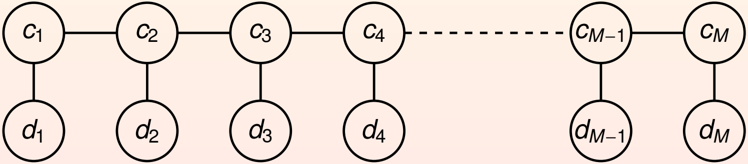
Context: Hidden Markov Model

Assign class $c_1, \ldots, c_M$ which maximizes

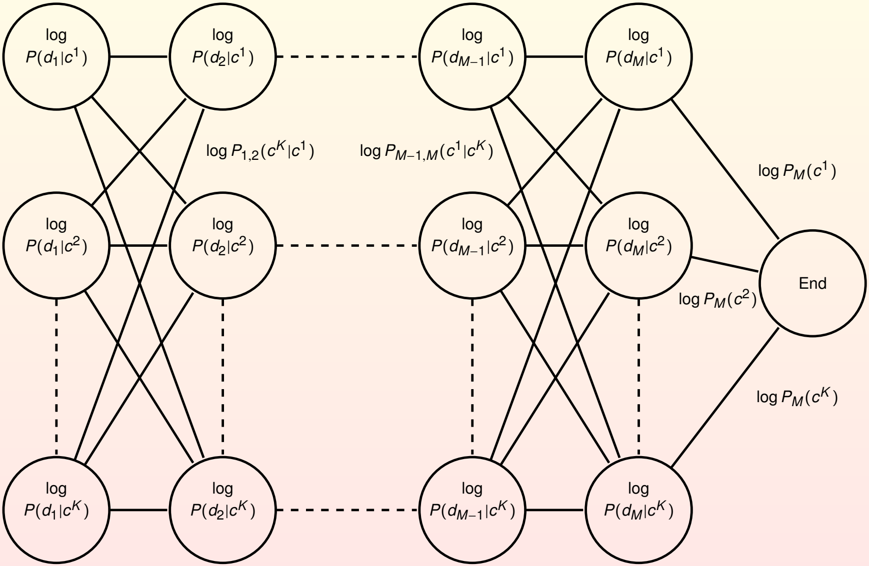
$$\prod_{m=1}^{M} P(d_m \mid c_m) P_{m,m+1}(c_m \mid c_{m+1}) \qquad (1)$$

Since log is a monotonically increasing function the class assignment $c_1, \ldots, c_M$ which maximizes (1) will maximize

$$\log P(c_1, \ldots, c_M, d_1, \ldots, d_M) = \sum_{m=1}^{M} \log P(d_m \mid c_m) + \log P_{m,m+1}(c_m \mid c_{m+1})$$



The Trellis



log $P(d_1|c^1)$
log $P(d_2|c^1)$
log $P(d_{M-1}|c^1)$
log $P(d_M|c^1)$
log $P_{1,2}(c^K|c^1)$
log $P_{M-1,M}(c^1|c^K)$
log $P_M(c^1)$
log $P(d_1|c^2)$
log $P(d_2|c^2)$
log $P(d_{M-1}|c^2)$
log $P(d_M|c^2)$
End
log $P_M(c^2)$
log $P_M(c^K)$
log $P(d_1|c^K)$
log $P(d_2|c^K)$
log $P(d_{M-1}|c^K)$
log $P(d_M|c^K)$

Viterbi

- $w(m+1,k)$ is the weight of the best path from column 1 to node k of column $m+1$
- $b(m+1,k)$ is the node of column m yielding the highest weight to node k of column $m+1$

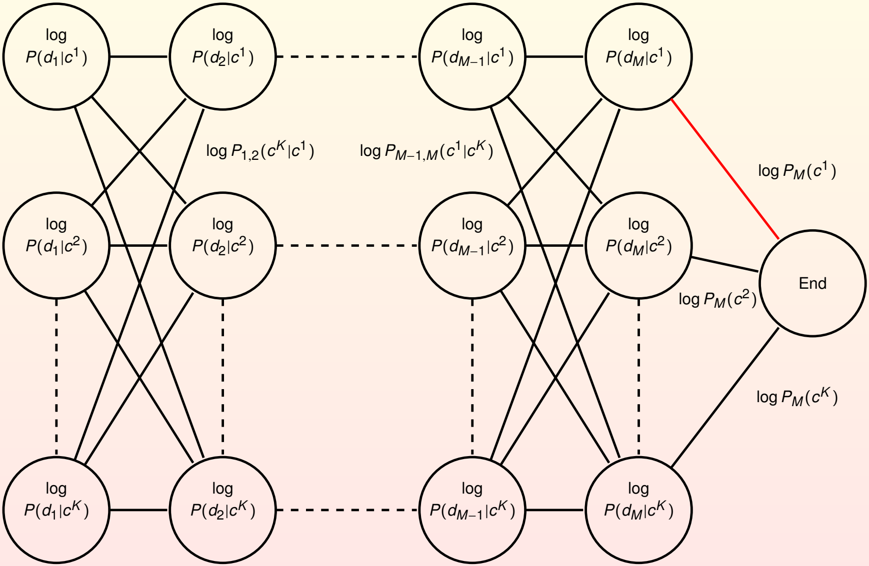
Initialization
for k=1,...,K

$$w(1,k) = \log P(d_1|c^k)$$

Column $m+1$ Calculation
for k=1,...,K

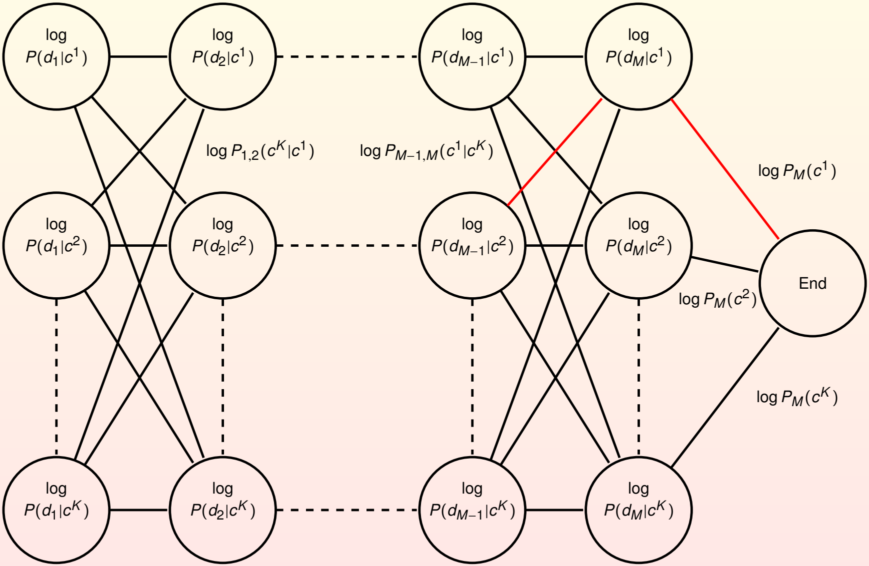
$$\begin{aligned} w(m+1,k) &= \log(P(d_{m+1}|c^k)) + \max_{\kappa=1,\ldots,K}[w(m,\kappa) + \log P_{m,m+1}(c^k|c^\kappa)] \\ b(m+1,k) &= \arg\max_{\kappa=1,\ldots,K}[w(m,\kappa) + \log P_{m,m+1}(c^k|c^\kappa)] \end{aligned}$$

The Trellis



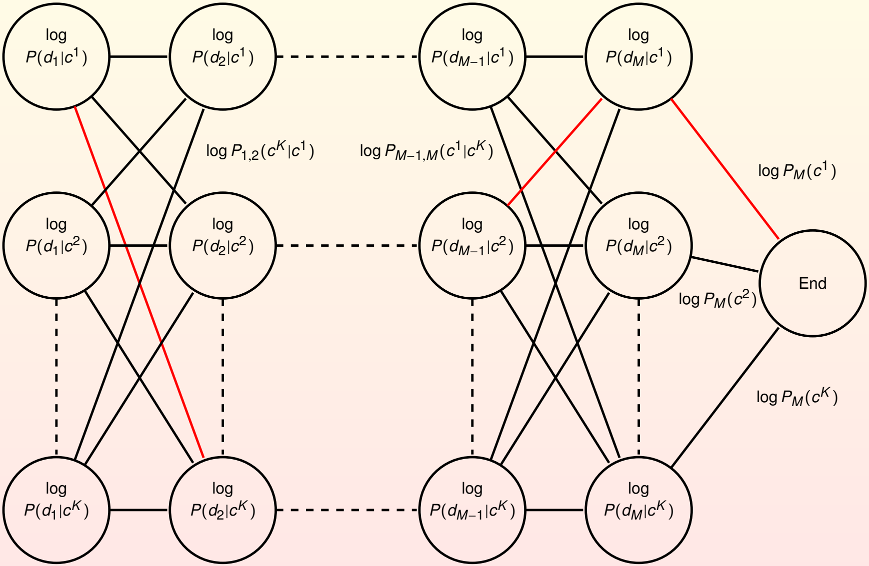
log $P(d_1|c^1)$
log $P(d_2|c^1)$
log $P(d_{M-1}|c^1)$
log $P(d_M|c^1)$
log $P_{1,2}(c^K|c^1)$
log $P_{M-1,M}(c^1|c^K)$
log $P_M(c^1)$
log $P(d_1|c^2)$
log $P(d_2|c^2)$
log $P(d_{M-1}|c^2)$
log $P(d_M|c^2)$
End
log $P_M(c^2)$
log $P_M(c^K)$
log $P(d_1|c^K)$
log $P(d_2|c^K)$
log $P(d_{M-1}|c^K)$
log $P(d_M|c^K)$

The Trellis



log $P(d_1|c^1)$
log $P(d_2|c^1)$
log $P(d_{M-1}|c^1)$
log $P(d_M|c^1)$
log $P_{1,2}(c^K|c^1)$
log $P_{M-1,M}(c^1|c^K)$
log $P_M(c^1)$
log $P(d_1|c^2)$
log $P(d_2|c^2)$
log $P(d_{M-1}|c^2)$
log $P(d_M|c^2)$
End
log $P_M(c^2)$
log $P_M(c^K)$
log $P(d_1|c^K)$
log $P(d_2|c^K)$
log $P(d_{M-1}|c^K)$
log $P(d_M|c^K)$

The Trellis


log $P(d_1|c^1)$
log $P(d_2|c^1)$
log $P(d_{M-1}|c^1)$
log $P(d_M|c^1)$
log $P_{1,2}(c^K|c^1)$
log $P_{M-1,M}(c^1|c^K)$
log $P_M(c^1)$
log $P(d_1|c^2)$
log $P(d_2|c^2)$
log $P(d_{M-1}|c^2)$
log $P(d_M|c^2)$
End
log $P_M(c^2)$
log $P_M(c^K)$
log $P(d_1|c^K)$
log $P(d_2|c^K)$
log $P(d_{M-1}|c^K)$
log $P(d_M|c^K)$

Viterbi Summary

The Viterbi algorithm determines the class sequence $c_1, \ldots, c_M$ that maximizes $P(c_1, \ldots, c_M, d_1, \ldots, d_M)$ assuming that $P(d_1, \ldots, d_M \mid c_1, \ldots, c_M) = \prod_{m=1}^{M} P(d_m \mid c_m)$ and that $P(c_1, \ldots, c_M) = \prod_{m-1}^{M} P_{m,m+1}(c_m \mid c_{m+1})$. The gain matrix that is consistent with this assignment methodology is

$$e(c_1^*, \ldots, c_M^*; c_1, \ldots, c_M) = \begin{cases} 1 & \text{if } c_1^* \ldots, c_M^* = c_1, \ldots, c_M \\ 0 & \text{otherwise} \end{cases}$$

Gain 1 if the assignment is completely correct; otherwise gain 0. There is no partial credit.

Viterbi Algorithm has the tendency to halucinate.

Bayes and Markov Processing System: BAMPS

What happens if the economic gain matrix is

$$e(c_1^*, \ldots, c_M^*; c_1, \ldots, c_M) = \sum_{m=1}^{M} e(c_m^*, c_m)$$

where $e(c^*, c) > 0$ when $c^* = c$ and 0 otherwise.
Assign $(c_1, \ldots, c_M)$ to maximize the expected gain

$$\begin{aligned} E[e] &= \sum_{(c_1^*,\ldots,c_M^*)} e(c_1^* \ldots, c_M^*; c_1, \ldots, c_M) P(c_1^* \ldots, c_M^*, d_1, \ldots, d_M) \\ &= \sum_{(c_1^*,\ldots,c_M^*)} \sum_{m=1}^{M} e(c_m^*, c_m) P(c_1^* \ldots, c_M^*, d_1, \ldots, d_M) \\ &= \sum_{m=1}^{M} \sum_{(c_1^*,\ldots,c_M^*)} e(c_m^*, c_m) P(c_1^* \ldots, c_M^*, d_1, \ldots, d_M) \end{aligned}$$

BAMPS: Forward Backwards Algorithm

Assign $(c_1, \ldots, c_M)$ to maximize the expected gain

$$
\begin{aligned}
E[e] &= \sum_{m=1}^{M} \sum_{(c_1^*, \ldots, c_M^*)} e(c_m^*, c_m) P(c_1^* \ldots, c_M^*, d_1, \ldots, d_M) \\
&= \sum_{m=1}^{M} \sum_{c_m^*} e(c_m^*, c_m) \sum_{c_1^*} \cdots \sum_{c_{m-1}^*} \sum_{c_{m+1}^*} \cdots \sum_{c_M} P(c_1^* \ldots, c_M^*, d_1, \ldots, d_M) \\
&= \sum_{m=1}^{M} \sum_{c_m^*} e(c_m^*, c_m) P(c_m^*, d_1, \ldots, d_M) \\
&= \sum_{m=1}^{M} e(c_m, c_m) P(c_m, d_1, \ldots, d_M)
\end{aligned}
$$

when the gain matrix is diagonal.

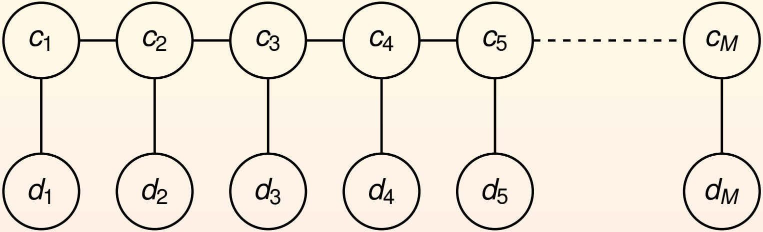
BAMPS: Forwards Backwards

When the gain matrix is diagonal, assign $c_1, \ldots, c_M$ to maximize

$$E[e] \;=\; \sum_{m=1}^{M} e(c_m, c_m) P(c_m, d_1, \ldots, d_M)$$

When the diagonal entries are positive, for each m, $m = 1, \ldots, M$, assign c_m to maximize $P(c_m, d_1, \ldots, d_M)$.

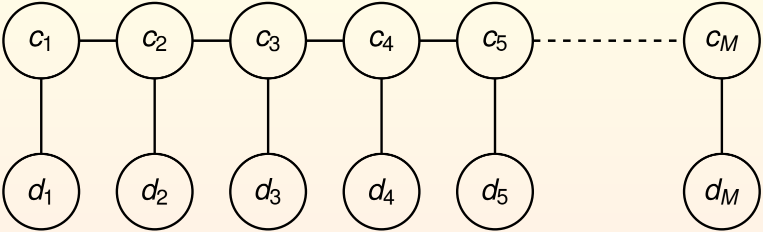
Conditional Independence Graph



Cliques: $\{d_1, c_1\}, \{c_1, c_2\}, \{c_2, d_2\}, \{c_2, c_3\}, \{c_3, d_3\} \ldots \{c_{M-1}, c_M\}, \{c_M, d_M\}$

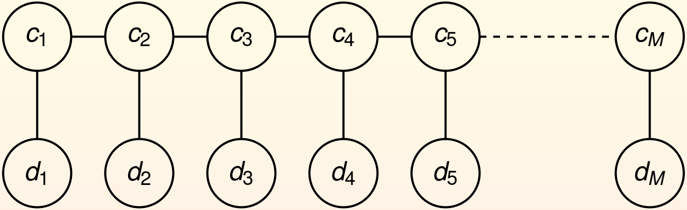
Separators $c_1, \quad c_2, \quad c_2, \quad c_3 \quad \ldots \quad c_{M-1}, \quad c_M$

Conditional Independence Graph



$$
\begin{aligned}
P(c_1, \ldots, c_M, d_1, \ldots, d_M) &= \frac{\prod_{i=1}^{M} P(d_i, c_i) \prod_{j=1}^{M-1} P(c_j, c_{j+1})}{\prod_{k=1}^{M} P(c_k) \prod_{m=2}^{M-1} P(c_m)} \\
&= \left[\prod_{i=1}^{M} P(d_i \mid c_i)\right] \left[\prod_{j=2}^{M} P(c_j \mid c_{j-1})\right] P(c_1)
\end{aligned}
$$

Conditional Independence Graph



$$\begin{aligned} P(c_1, \ldots, c_M) &= \frac{\prod_{j=1}^{M-1} P(c_j, c_{j+1})}{\prod_{m=2}^{M-1} P(c_m)} = \frac{\prod_{i=1}^{m-1} P(c_i, c_{i+1})}{\prod_{j=2}^{m-1} P(c_j)} \frac{\prod_{k=m}^{M-1} P(c_k, c_{k+1})}{\prod_{n=m}^{M-1} P(c_n)} \\ &= P(c_1) \frac{\prod_{i=1}^{m-1} P(c_i, c_{i+1})}{\prod_{j=1}^{m-1} P(c_j)} \frac{\prod_{k=m}^{M-1} P(c_k, c_{k+1})}{P(c_m) \prod_{n=m}^{M-1} P(c_{n+1})} P(c_M) \\ &= P(c_1) \left[\prod_{i=1}^{m-1} P(c_{i+1} \mid c_i) \right] \frac{1}{P(c_m)} \left[\prod_{k=m}^{M-1} P(c_k \mid c_{k+1}) \right] P(c_M) \end{aligned}$$

BAMPS; Forwards Backwards

When the diagonal entries are positive, for each m, $m = 1, \ldots, M$, assign c_m to maximize $P(c_m, d_1, \ldots, d_M)$. For $2 \le m \le M-1$

$$\begin{aligned}
P(c_m, d_1, \ldots, d_M) &= \sum_{c_1} \cdots \sum_{c_{m-1}} \sum_{c_{m+1}} \cdots \sum_{c_M} P(c_1, \ldots, c_M, d_1, \ldots, d_M) \\
&= \sum_{c_1} \cdots \sum_{c_{m-1}} \sum_{c_{m+1}} \cdots \sum_{c_M} \prod_{i=1}^{M} P(d_i \mid c_i) \\
&\qquad P(c_1) \prod_{i=1}^{m-1} P(c_{i+1} \mid c_i) \frac{1}{P(c_m)} \prod_{k=m+1}^{M} P(c_{k-1} \mid c_k) P(c_M) \\
&= \left[\sum_{c_1} P(c_1) \cdots \sum_{c_{m-1}} \prod_{i=1}^{m-1} P(d_i \mid c_i) P(c_{i+1} \mid c_i) \right] \frac{P(d_m \mid c_m)}{P(c_m)} \\
&\qquad \left[\sum_{c_{m+1}} \cdots \sum_{c_M} P(c_M) \prod_{j=m+1}^{M} P(d_j \mid c_j) P(c_{j-1} \mid c_j) \right] \\
&= \left[\sum_{c_{m-1}} P(d_{m-1} \mid c_{m-1}) P(c_m \mid c_{m-1}) \cdots \sum_{c_1} P(d_1 \mid c_1) P(c_2 \mid c_1) P(c_1) \right] \frac{P(d_m \mid c_m)}{P(c_m)} \\
&\qquad \left[\sum_{c_{m+1}} P(d_{m+1} \mid c_{m+1}) P(c_m \mid c_{m+1}) \cdots \sum_{c_M} P(d_M \mid c_M) P(c_{M-1} \mid c_M) P(c_M) \right]
\end{aligned}$$

BAMPS: Forwards Backwards

$$P(c_m, d_1, \ldots, d_M) = \left[\sum_{c_{m-1}} P(d_{m-1} \mid c_{m-1}) P(c_m \mid c_{m-1}) \cdots \sum_{c_1} P(d_1 \mid c_1) P(c_2 \mid c_1) P(c_1)\right] \frac{P(d_m \mid c_m)}{P(c_m)}$$

$$\left[\sum_{c_{m+1}} P(d_{m+1} \mid c_{m+1}) P(c_m \mid c_{m+1}) \cdots \sum_{c_M} P(d_M \mid c_M) P(c_{M-1} \mid c_M) P(c_M)\right]$$

Define

$$f_m(c_m) = \sum_{c_{m-1}} P(d_{m-1} \mid c_{m-1}) P(c_m \mid c_{m-1}) \cdots \sum_{c_1} P(d_1 \mid c_1) P(c_2 \mid c_1) P(c_1), \; m = 2, \ldots M$$

$$b_m(c_m) = \sum_{c_{m+1}} P(d_{m+1} \mid c_{m+1}) P(c_m \mid c_{m+1}) \cdots \sum_{c_M} P(d_M \mid c_M) P(c_{M-1} \mid c_M) P(c_M)$$

$$m = M - 1 \ldots, 1$$

Then

$$P(c_m, d_1, \ldots, d_M) = f_m(c_m) \frac{P(d_m \mid c_m)}{P(c_m)} b_m(c_m), \; m = 1, \ldots, M$$

Where

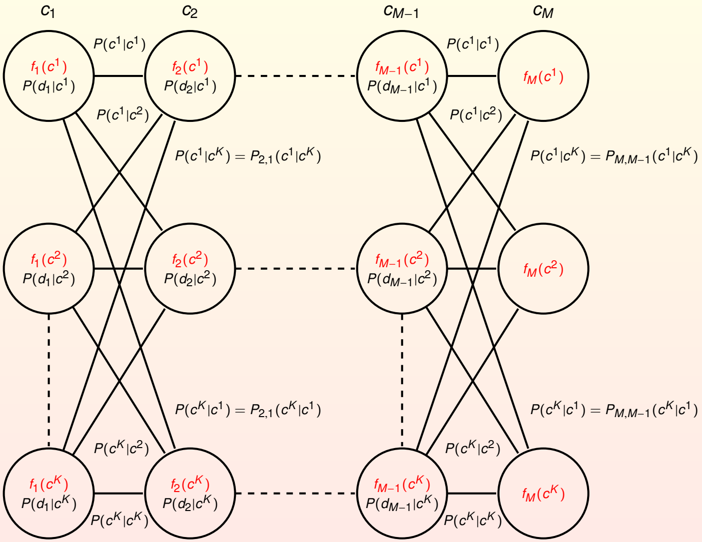
$$f_1(c_1) = P(c_1)$$

$$f_k(c_k) = \sum_{c_{k-1}} P(d_{k-1} \mid c_{k-1}) P_{k,k-1}(c_k \mid c_{k-1}) f_{k-1}(c_{k-1}), \; k = 2, \ldots, M$$

$$b_M(c_M) = P(c_M)$$

$$b_k(c_k) = \sum_{c_{k+1}} P(d_{k+1} \mid c_{k+1}) P_{k,k+1}(c_k \mid c_{k+1}) b_{k+1}(c_{k+1}), \; k = M - 1, \ldots, 1$$

The Trellis



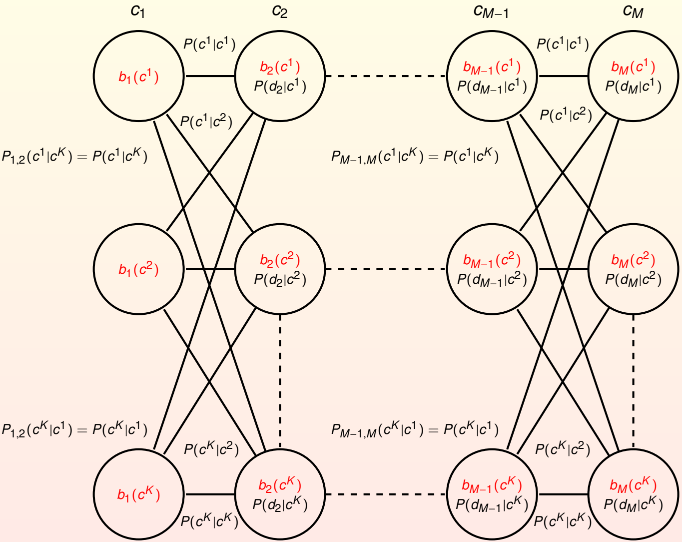
$f_1(c^k) = P(c^k)$

$k = 1, \ldots, K$

$C = \{c^1, \ldots, c^K\}$

$$f_m(c) \quad = \quad \sum_{c^* \in C} P(d_{m-1} \mid c^*) P_{m,m-1}(c \mid c^*) f_{m-1}(c^*), \; m = 2, \ldots, M; \; c \in C$$

The Trellis



$b_M(c^k) = P(c^k)$

$k = 1, \ldots, K$

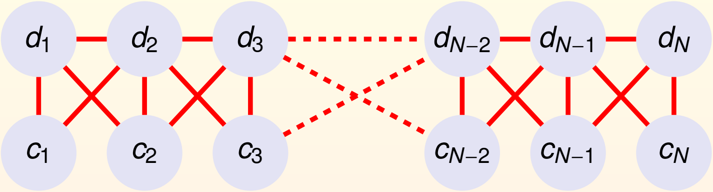
$C = \{c^1, \ldots, c^K\}$

$$b_m(c) = \sum_{c^* \in C} P(d_{m+1} \mid c^*) P_{m,m+1}(c \mid c^*) b_{m+1}(c^*), \; m = M-1, \ldots, 1; \; c \in C$$

Using the Graphical Model

- Let X be the set of all random variables
- Let $A, B, C \subseteq X$
- Let $M(A)$ be the Markov Blanket of A
- $P(A \mid X - A) = P(A \mid M(A))$
  - Because the Markov Blanket of A separates A from $X - A - M(A)$
- If all paths between A and B go through C, then $P(AB \mid C) = P(A \mid C)P(B \mid C)$
  - Because C separates A from B

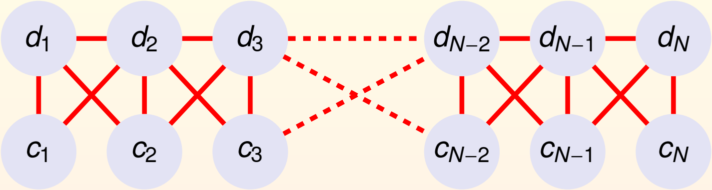
Pattern Dependent Graphical Model



The Markov blanket (or boundary) of node c_n is d_{n-1}, d_n, d_{n+1}. Therefore, c_n is conditionally independent of $c_1, \ldots, c_{n-1}, c_{n+1} \ldots, c_N, d_1, \ldots, d_{n-2}, d_{n+2}, \ldots, d_N$ given d_{n-1}, d_n, d_{n+1}.

$$
\begin{aligned}
P(c_n \quad | \quad & c_1, \ldots, c_{n-1}, c_{n+1} \ldots, c_N, d_1, \ldots, \ldots, d_N) \\
= \quad & P(c_n \mid d_1, \ldots, d_N) \\
= \quad & P(c_n \mid d_{n-1}, d_n, d_{n+1})
\end{aligned}
$$

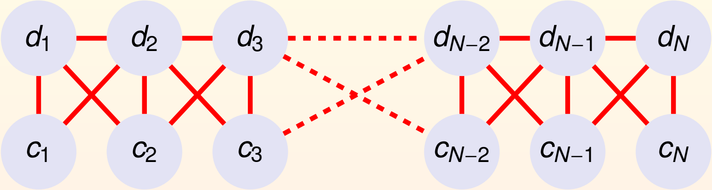
Patten Dependent Graphical Model



The Markov blanket of node d_n is $d_{n-1} d_{n+1}, c_{n-1}, c_n, c_{n+1}$. Therefore d_n is conditionally independent of $c_1, \ldots, c_N, d_1, \ldots, d_{n-2}, d_{n+2}, \ldots, d_N$ given $d_{n-1} d_{n+1}, c_{n-1}, c_n, c_{n+1}$.

$$\begin{aligned} P(d_n \quad &| \quad c_1, \ldots, c_N, d_1, \ldots, d_{n-2}, d_{n+2}, \ldots, d_N) \\ &= \quad P(d_n \mid d_{n-1}, d_{n+1}, c_{n-1}, c_n, c_{n+1}) \end{aligned}$$

Patten Dependent Graphical Model



All paths between nodes d_{n-1} and d_{n+1} must go through one of the nodes d_n or c_n. This means that d_{n-1} is conditionally independent of d_{n+1} given d_n and c_n.

$$P(d_{n-1}, d_{n+1} \mid d_n, c_n) = P(d_{n-1} \mid d_n, c_n) P(d_{n+1} \mid d_n, c_n)$$

Pattern Dependent Graphical Model

For $n = 2, \ldots, N-1$

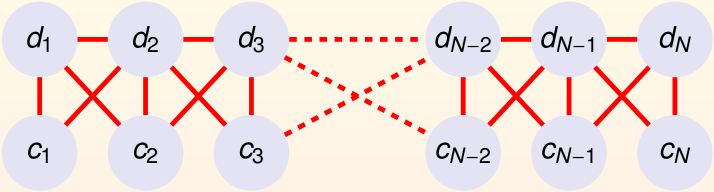
$$\begin{aligned} P(c_n | d_{n-1}, d_n, d_{n+1}) &= \frac{P(d_{n-1}, d_n, d_{n+1}, c_n)}{P(d_{n-1}, d_n, d_{n+1})} \\ &= \frac{P(d_{n-1}, d_{n+1} | d_n, c_n) P(d_n, c_n)}{P(d_{n-1}, d_n, d_{n+1})} \\ &= \frac{P(d_{n-1} | d_n, c_n) P(d_{n+1} | d_n, c_n) P(d_n | c_n) P(c_n)}{P(d_{n-1}, d_n, d_{n+1})} \end{aligned}$$

Pattern Dependent Graphical Model

Putting these conditional independences together, for $n = 2, \ldots, N-1$

$$\begin{aligned} P(c_n|d_1,\ldots,d_N) &= P(c_n|d_{n-1},d_n,d_{n+1}) \\ &= \frac{P(d_{n-1}|d_n,c_n)P(d_{n+1}|d_n,c_n)P(d_n|c_n)P(c_n)}{P(d_{n-1},d_n,d_{n+1})} \end{aligned}$$

Cliques and Separators



There are $2N-2$ cliques and $2N-3$ separators.

$$\{d_1, d_2, c_1\}, \{d_1, d_2, c_2\}, \{d_2, d_3, c_2\}, \{d_2, d_3, c_3\}, \ldots \{d_{N-1}, d_N, c_n\}$$
$$\{d_1, d_2\}, \quad \{d_2, c_2\}, \quad \{d_2, d_3\}, \ldots, \{d_{N-1}, d_N\}$$

Cliques and Separators

$$P(c_1,\ldots,c_N,d_1,\ldots,d_N) \quad = \quad P(d_2,d_1,c_1)\left(\prod_{n=2}^{N-1}\frac{P(d_{n-1},d_n,c_n)P(d_{n+1}d_n,c_n)}{P(d_{n-1},d_n)P(d_n,c_n)}\right)\frac{P(d_{N-1},d_N,c_N)}{P(d_{N-1},d_N)}$$

$$P(c_1,\ldots,c_N|d_1,\ldots,d_N) \quad = \quad \frac{\prod_{n=1}^{N}P(d_{n-1}|d_n,c_n)P(d_{n+1}|d_n,c_n)P(d_n|c_n)P(c_n)}{\sum_{c_1,\ldots,c_N}\prod_{k=1}^{N}P(d_{k-1}|d_k,c_k)P(d_{k+1}|d_k,c_k)P(d_k|c_k)P(c_k)}$$

Generalized Products

Definition

Let $B_1, \ldots, B_K$ be index subsets of {1,...,N}. The product form $\prod_{k=1}^{K} a_k(x_i : i \in B_k)$ is called a *generalized product form* if and only if for some probability function $P(x_1, \ldots, x_N)$

- $P(x_1, \ldots, x_N) = \prod_{k=1}^{K} a_k(x_i : i \in B_k)$
- $P(x_1, \ldots, x_N)$ is an extension of $P(x_i : i \in B_k), k = 1, \ldots, K$

Generalized Products

Let $B_1, \ldots, B_K$ be index subsets of {1,. . . ,N}. Given marginal probability functions $P(x_i : i \in B_k), k = 1, \ldots, K$ find functions $a_k(x_i : i \in B_k)$ such that

- $P(x_1, \ldots, x_N) = \prod_{k=1}^{K} a_k(x_i : i \in B_k)$
- $P(x_1, \ldots, x_N)$ is an extension of $P(x_i : i \in B_k), k = 1, \ldots, K$

Projections

Definition

Let $S = \{s_1, \ldots, s_M\}$ be an index subset of $\{1, \ldots, N\}$. $\pi_S(x_1, \ldots, x_N)$ is called the *projection* of $(x_1, \ldots, x_N)$ onto the index set S. $\pi_S(x) = (x_{s_1}, \ldots, x_{s_M}) = (x_i : i \in S)$.

If $(x_1, x_2, x_3, x_4, x_5) = (1, 5, 4, 3, 0)$ and $S = \{1, 4, 5\}$, then $\pi_S(1, 5, 4, 3, 0) = (x_i : i \in S) = (1, 3, 0)$.

Inverse Projection

Definition

Let h be a tuple whose components are indexed in index set S. Let I be the index set for all the variables. The *inverse projection* $\pi_I^{-1} h$ of h with respect to I is defined by

$$\pi_I^{-1}(h) = \{(x_1, \ldots, x_N) \mid \pi_S(x_1, \ldots, x_N) = h\}$$

Marginals

Let P be a probability function on N variables $(x_1, \ldots, x_N)$. Let $S_0, \ldots, S_{K-1}$ be K index sets of $\{1, \ldots, N\}$ covering $\{1, \ldots, N\}$. Fix k. Let h be a tuple whose components are indexed in index set S_k: $h = (x_i : i \in S_k)$.

$$P(h) = P(x_i : i \in S_k) = \sum_{(x_1,\ldots,x_N) \in \pi_I^{-1}(h)} P(x_1, \ldots, x_N)$$

Extension

Definition

Let P be a probability function on N variables $(x_1, \ldots, x_N)$. Let $S_0, \ldots, S_{K-1}$ be K index sets of $I = \{1, \ldots, N\}$. Let f_k be marginal probability functions defined on tuples $h_k = (x_i : i \in S_k)$, $k = 0, \ldots, K-1$. P is an *extension* of marginals $f_1, \ldots, f_K$ if and only if

$$f_k(h_k) = \sum_{(x_1,\ldots,x_N)\in\pi_I^{-1}(h_k)} P(x_1, \ldots, x_N)$$