

# Probabilistic Models for Occurrences

Modelling probabilistic dependencies (causalities) and independencies between discrete events of a scene

$X_i$  random variable *models uncertain propositions about a scene*

$X_i = a$  hypothesis

Decomposition of joint probabilities:

$$P(X_1, X_2, X_3, \dots, X_N) = P(X_1 | X_2, X_3, \dots, X_N) \cdot P(X_2 | X_3, X_4, \dots, X_N) \cdot \dots \cdot P(X_{N-1} | X_N) \cdot P(X_N)$$

Simplification in the case of statistical independence:

X independent of  $X_i$

$$P(X | X_1, \dots, X_{i-1}, X_i, X_{i+1}, \dots, X_N) = P(X | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_N)$$

Joint probability of N variables may be simplified by ordering the variables according to their direct dependence (causality).

Note notation:  $P(X_1, \dots, X_N) = \dots$  means  $\forall x_1 \dots \forall x_N P(X_1 = x_1, \dots, X_N = x_N) = \dots$

# Independence Causes Complexity Reduction

Assume that all random variables  $X_n$  of the JPD  $P(X_1, X_2, X_3, \dots, X_N)$  have a domain size  $K$ . Then a fully general JPD requires  $K^N$  entries.

Example:  $N = 20, K = 10 \Rightarrow 10^{20}$  entries must be specified!

If all random variables are statistically independent, we have  $P(X_1, X_2, X_3, \dots, X_N) = P(X_1) \cdot P(X_2) \cdot \dots \cdot P(X_N)$  and only  $KN$  entries are required.

**Exploiting independencies can greatly reduce the size of a probability table!**

# Conditional Independence

It is useful to determine direct influences  $Y_i$  on a random variable  $X$ , because given the  $Y_i$ ,  $X$  is independent of other Variables  $Z_k$  "upstream" to the  $Y_i$ .

Let  $\text{dom}(X)$  be the domain of  $X$ , i.e. the set of possible values of  $X$ .

A random variable  $X$  is independent of  $Z$  given  $Y$  if for all  $x_i \in \text{dom}(X)$ , for all  $y_j \in \text{dom}(Y)$ , and for all  $z_k \in \text{dom}(Z)$ ,

$$P(X=x_i \mid Y=y_j, Z=z_k) = P(X=x_i \mid Y=y_j)$$

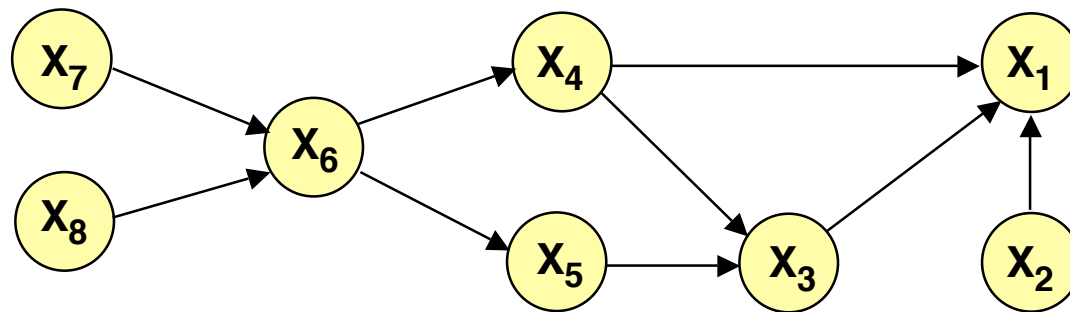
Example:  $X=\text{plate\_on\_table}$ ,  $Y=\text{laying\_table}$ ,  $Z=\text{want\_to\_eat}$

X Y Z	P(XYZ)	X Y Z	P(XYZ)	Check whether X is independent of Z given Y!
T T T	.096	F T T	.024	
T T F	.064	F T F	.016	
T F T	.0	F F T	.08	
T F F	.0	F F F	.72	

# Causality Graph

Conditional dependencies (causality relations) of random variables define partial order.

Representation as a directed acyclic graph (DAG):



$$P(X_1, X_2, X_3, \dots, X_8) =$$

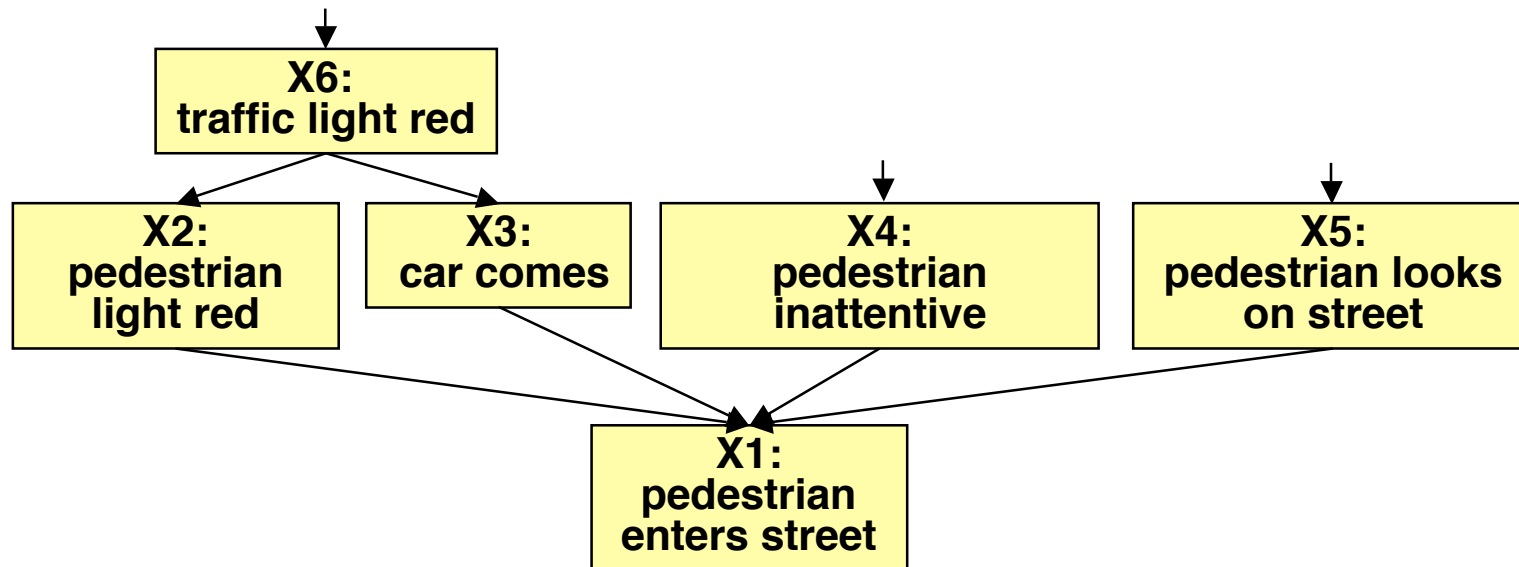
$$P(X_1 | X_2, X_3, X_4) \cdot P(X_2) \cdot P(X_3 | X_4, X_5) \cdot P(X_4 | X_6) \cdot P(X_5 | X_6) \cdot P(X_6 | X_7 X_8) \cdot P(X_7) \cdot P(X_8)$$

For any DAG, we obtain the JPD as follows:

$\text{Pa}(X_i)$  parents of node  $X_i$

$$P(X_1 \dots X_N) = \prod_i P(X_i | \text{Pa}(X_i))$$

# Example: Traffic Behaviour of Pedestrians



Conditional probability table for each node must be known

$P(X1 \mid X2, X3, X4, X5)$

X1	X2	X3	X4	X5	P
T	T	T	T	T	0.3
F	T	T	T	T	0.7
T	F	T	T	T	0.9
F	F	T	T	T	0.1
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:

$P(X2 \mid X6)$

X2	X6	P
T	T	0.2
F	T	0.8
T	F	1.0
F	F	0.0

$P(X3 \mid X6)$

X3	X6	P
T	T	0.01
F	T	0.99
T	F	0.6
F	F	0.4

$P(X4)$

X4	P
T	0.1
F	0.9

$P(X5)$

X5	P
T	0.7
F	0.3

$P(X6)$

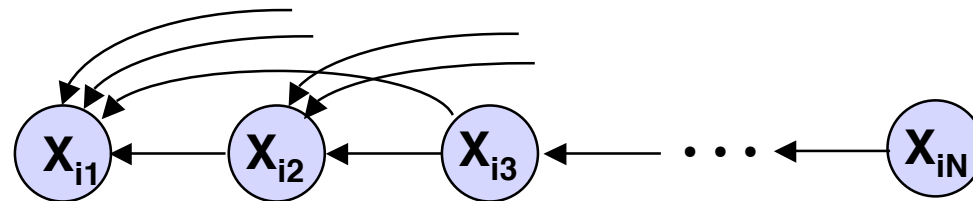
X6	P
T	0.7
F	0.3

# Bayes Nets are not Unique

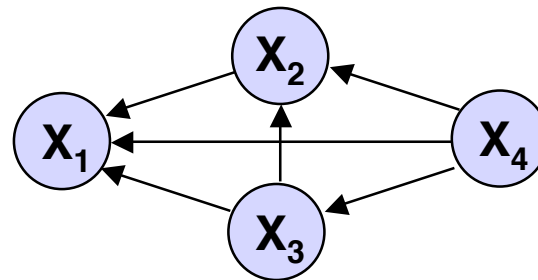
Using the chain rule, a JPD  $P(X_1, X_2, \dots, X_N)$  may be expanded in  $N!$  ways:

$$P(X_1, X_2, \dots, X_N) = P(X_{i_1} | X_{i_2}, \dots, X_{i_N}) \cdot P(X_{i_2} | X_{i_3}, \dots, X_{i_N}) \cdot \dots \cdot P(X_{i_N})$$

Even with no independencies, each chain rule expansion can be drawn as a graphical model:



Example:



Any JPD  $P(X_1, X_2, X_3, X_4)$  can be represented by this Bayes Net.

For efficient inferences with a given JPD, it is important to find a Bayes Net with a low number of dependencies.

# Constructing a Bayes Net

## By domain analysis:

1. Select discrete variables  $X_i$  relevant for domain
2. Establish partial order of variables according to causality
3. In the order of decreasing causality:
  - (i) Generate node  $X_i$  in net
  - (ii) As predecessors of  $X_i$  choose the smallest subset of nodes which are already in the net and from which  $X_i$  is causally dependent
  - (iii) Determine a table of conditional probabilities for  $X_i$

## By data analysis:

Use a learning method to establish a Bayes Net approximating the empirical joint probability distribution.

# Computing Inferences

We want to use a Bayes Net for probabilistic inferences of the following kind:

Given a joint probability  $P(X_1, \dots, X_N)$  represented by a Bayes Net, and evidence  $X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}$  for some of the variables, what is the probability  $P(X_n = a_i \mid X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})$  of an unobserved variable to take on a value  $a_i$  ?

In general this requires

- expressing a conditional probability by a quotient of joint probabilities

$$P(X_n = a_i \mid X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = \frac{P(X_n = a_i, X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})}{P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})}$$

- determining partial joint probabilities from the given total joint probability by summing out unwanted variables

$$P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = \sum_{X_{n_1}, \dots, X_{n_K}} P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}, X_{n_1}, \dots, X_{n_K})$$



# Normalization

Basic formula for computing the probability of a query variable  $X_n$  from a JPD  $P(X_1, \dots, X_N)$  given evidence  $X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}$  :

$$P(X_n = a_i \mid X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = \frac{P(X_n = a_i, X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})}{P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})}$$

The denominator on the right is independent of  $a_i$  and constitutes a normalizing factor  $\alpha$ . It can be computed by requiring that the conditional probabilities of all  $a_i$  sum to unity.

$$P(X_n = a_i \mid X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = \alpha \{ P(X_n = a_i, X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) \}$$

Formulae are often written in this simplified form with  $\alpha$  as a normalizing factor.

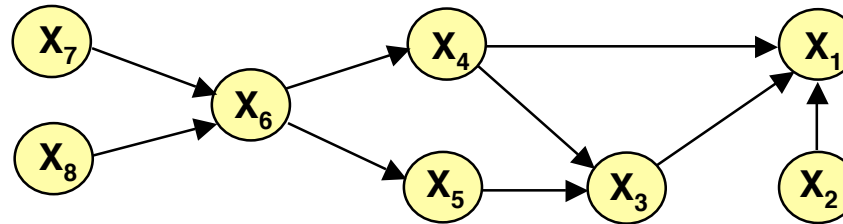
# Factoring the JPD

JPDs can be computed from a Bayes Net more efficiently by ordering the "factors" so that only few summations and products must be computed.

Example:

Compute

$$P(X_2=a, X_4=b \mid X_1=c, X_7=d)$$



$$P(X_2=a, X_4=b \mid X_1=c, X_7=d) = \frac{P(X_2=a, X_4=b, X_1=c, X_7=d)}{P(X_1=c, X_7=d)}$$

$$P(X_2=a, X_4=b, X_1=c, X_7=d) = \sum_{X_3} \sum_{X_5} \sum_{X_6} \sum_{X_8} P(X_1=c, X_2=a, X_3, X_4=b, X_5, X_6, X_7=d, X_8)$$

$$= \sum_{X_3} \sum_{X_5} \sum_{X_6} \sum_{X_8} P(X_1=c \mid X_2=a, X_3, X_4=b) \cdot P(X_2=a) \cdot P(X_3 \mid X_4=b, X_5) \cdot P(X_4=b \mid X_6) \cdot P(X_5 \mid X_6) \cdot P(X_6 \mid X_7=d, X_8) \cdot P(X_7=d) \cdot P(X_8)$$

$$= P(X_2=a) \cdot P(X_7=d) \cdot \sum_{X_3} P(X_1=c \mid X_2=a, X_3, X_4=b) \cdot \sum_{X_5} P(X_3 \mid X_4=b, X_5) \cdot \sum_{X_6} P(X_4=b \mid X_6) \cdot P(X_5 \mid X_6) \cdot \sum_{X_8} P(X_6 \mid X_7=d, X_8) \cdot P(X_8)$$

**one possible  
order for  
efficient  
computation**

# Set-factoring Heuristic

Finding the best possible order for computing factors of a JPD is not tractable, in general. The set-factoring heuristic is a greedy (suboptimal) algorithm with often excellent results.

Given  $\mathcal{X}$  set of random variables to be summed out

$\mathcal{F}$  set of factors to be combined

**Set-factoring heuristic:**

- Pick the pair of factors which produces the smallest probability table after combination and summing out as many variables of  $\mathcal{X}$  as possible. Break ties by choosing the pair where most variables are summed out.
- Place resulting factor into set  $\mathcal{F}$ , remove summed-out variables from  $\mathcal{X}$  and repeat procedure.

# Example for Set-factoring Heuristic (1)

Compute

$$P(X_2=a, X_4=b, X_1=c, X_7=d) = \sum_{X_3} \sum_{X_5} \sum_{X_6} \sum_{X_8} P(X_1=c | X_2=a, X_3, X_4=b) \cdot P(X_2=a) \cdot P(X_3 | X_4=b, X_5) \cdot \\ \cdot P(X_4=b | X_6) \cdot P(X_5 | X_6) \cdot P(X_6 | X_7=d, X_8) \cdot P(X_7=d) \cdot P(X_8)$$

**Step 1:**  $\mathcal{X} = \{X_3, X_5, X_6, X_8\}$

$$\mathcal{F} = \{P(X_1=c | X_2=a, X_3, X_4=b), P(X_2=a), P(X_3 | X_4=b, X_5), P(X_4=b | X_6), P(X_5 | X_6), \\ P(X_6 | X_7=d, X_8), P(X_7=d), P(X_8)\}$$

After extracting the constant factors  $P(X_2=a)$  and  $P(X_7=d)$ , 6 factors remain, hence 15 possible pairs may be formed. Assuming equally sized domains, the set-factoring heuristic prefers 2 combinations:

- (i)  $P(X_1=c | X_2=a, X_3, X_4=b) \cdot P(X_3 | X_4=b, X_5)$  and summing out  $X_3$
- (ii)  $P(X_6 | X_7=d, X_8) \cdot P(X_8)$  and summing out  $X_8$

Choosing (ii), the new factor  $P(X_6 | X_7=d)$  is computed and the sets are updated:

**Step 2:**  $\mathcal{X} = \{X_3, X_5, X_6\}$

$$\mathcal{F} = \{P(X_1=c | X_2=a, X_3, X_4=b), P(X_3 | X_4=b, X_5), P(X_4=b | X_6), P(X_5 | X_6), P(X_6 | X_7=d)\}$$

## Example for Set-factoring Heuristic (2)

The set-factoring heuristic prefers the combination:

$$P(X_1=c | X_2=a, X_3, X_4=b) \cdot P(X_3 | X_4=b, X_5) \text{ and summing out } X_3$$

The new factor  $P(X_1=c | X_2=a, X_4=b, X_5)$  is computed and the sets are updated:

**Step 3:**  $\mathcal{X} = \{X_5, X_6\}$

$$\mathcal{F} = \{P(X_1=c | X_2=a, X_4=b, X_5), P(X_4=b | X_6), P(X_5 | X_6), P(X_6 | X_7=d)\}$$

The set-factoring heuristic prefers the combination:

$$P(X_1=c | X_2=a, X_4=b, X_5) \cdot P(X_5 | X_6) \text{ and summing out } X_5$$

The new factor  $P(X_1=c | X_2=a, X_4=b, X_6)$  is computed and the sets are updated:

**Step 4:**  $\mathcal{X} = \{X_6\}$

$$\mathcal{F} = \{P(X_1=c | X_2=a, X_4=b, X_6), P(X_4=b | X_6), P(X_6 | X_7=d)\}$$

The set-factoring heuristic ranks all combinations equal. Choosing

$$P(X_4=b | X_6) \cdot P(X_6 | X_7=d)$$

we get the new factor  $P(X_4=b, X_6 | X_7=d)$  and the updated sets:

**Step 5:**  $\mathcal{X} = \{X_6\}$

$$\mathcal{F} = \{P(X_1=c | X_2=a, X_4=b, X_6), P(X_4=b, X_6 | X_7=d)\}$$

## Example for Set-factoring Heuristic (3)

The final result follows from reassembling the summations outwards:

$$P(X_2=a, X_4=b, X_1=c, X_7=d) =$$

$$P(X_2=a) \cdot P(X_7=d)$$

$$\cdot \sum_{X_6} P(X_4=b | X_6) \cdot P(X_6 | X_7=d)$$

$$\cdot \sum_{X_5} P(X_5 | X_6)$$

$$\cdot \sum_{X_3} P(X_1=c | X_2=a, X_3, X_4=b) \cdot P(X_3 | X_4=b, X_5)$$

$$\cdot \sum_{X_8} P(X_6 | X_7=d, X_8) \cdot P(X_8)$$

If  $D$  is the size of the domains of the random variables, the number of multiplications is

$$N_{\text{mult}} = D^2 + D^2 + D^2 + D$$

This is the same as the number of multiplications for the manual ordering proposed earlier:

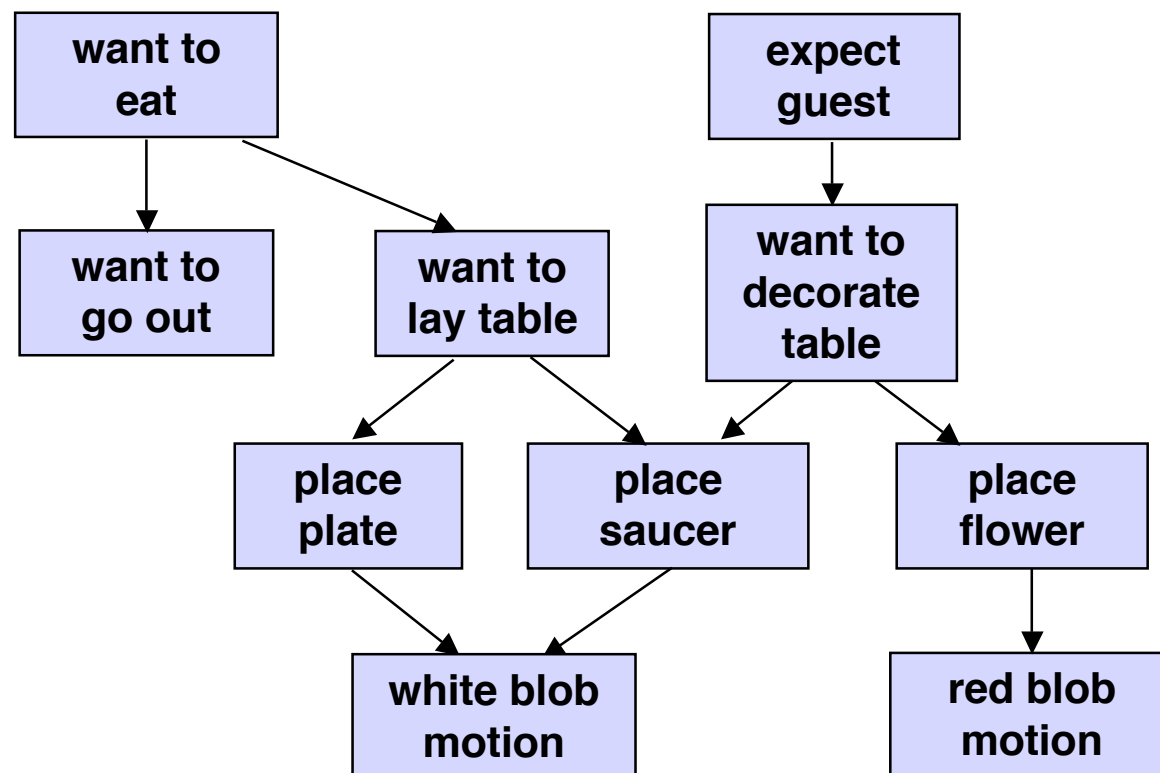
$$N_{\text{mult}} = D^2 + D^2 + D^2 + D$$

In this case, the heuristic did not reduce the computational expenses.

# Dependance Analysis of Bayes Nets

The arcs in a Bayes Net indicate pairwise independence. Can one infer other independencies

- in general?
- given partial evidence in terms of node values?



## Example:

If it is known

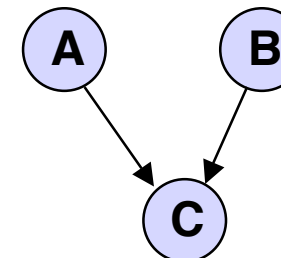
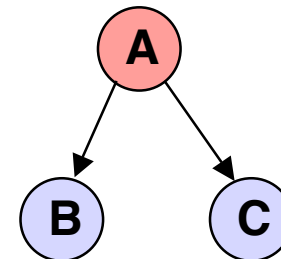
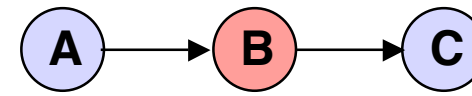
- that one wants to lay the table and
- that a white blob motion has been observed, does this affect the probability of
- wanting to go out?
- red blob motion?

# Blocking Evidence

In general, (undirected) paths in a Bayes Net indicate possible flow of information. However, if hard evidence is given at an intermediate node, the path may be blocked.

**Blocking situations:**

1. In a serial connection from A to C via B, evidence from A to C is blocked by hard evidence about B.
2. In a diverging connection from A to B and C, evidence from B to C is blocked by hard evidence about A.
3. In a converging situation from A and B to C, any evidence about C results in evidence transmitted between A and B.



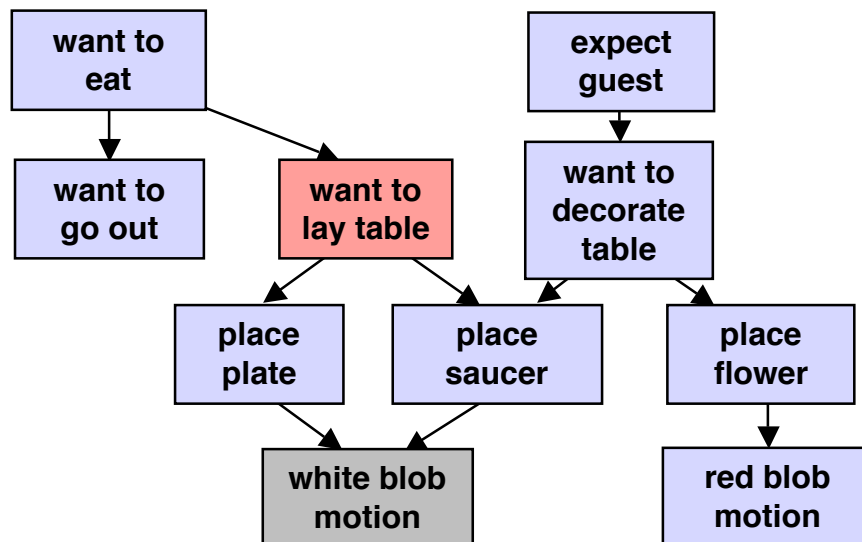


# D-separation

"D-separation" = no flow of evidence from one node to another

Two nodes X and Y in a Bayes Net are d-separated if, for all paths between X and Y, there is an intermediate node Z for which either:

1. the connection is serial or diverging and the value of Z is known for certain; or
2. the connection is converging and neither Z (nor any of its descendants) have received any evidence at all.



## Example:

Hard evidence for "want to lay table" blocks influence of evidence for "white blob motion" on "want to eat" and "want to go out", but not on any other nodes.

# Basic Kinds of Inferences

## 1. Causal reasoning, prediction

Given upstream evidence, ask for downstream probability

Example: Given "want to eat" is true, what is the probability of "white blob motion"?

## 2. Evidential reasoning, explanation

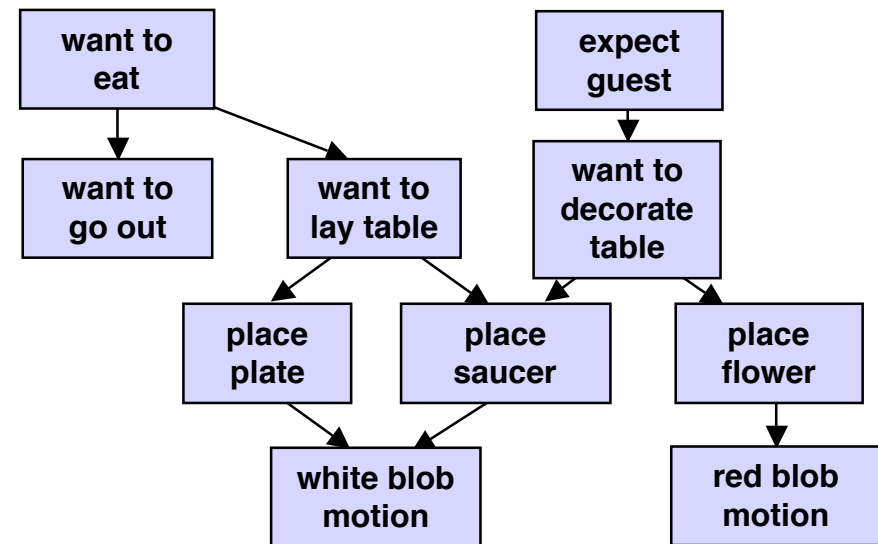
Given downstream evidence, ask for upstream probability

Example: Given "white blob motion" is true, what is the probability of "expect guest"?

## 3. Explaining away

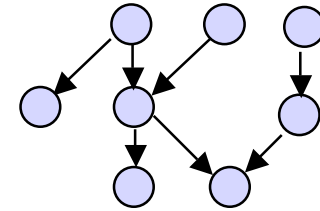
Given evidence of a node with two parents and evidence for one of the parents, ask for probability of other parent node

Example: Given evidence for "place saucer" and "want to eat", what is the probability of "want to decorate table"?

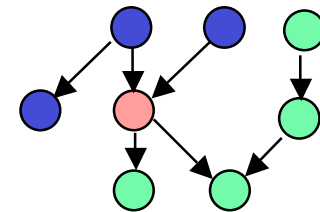


# Evidence Propagation in Polytrees

polytree = DAG where each pair of distinct nodes is connected by a single (undirected) path



Any node  $X_k$  in a polytree separates the tree into an "upper" and "lower" part. Hence the marginal probability  $P(X_k=c)$  can be computed from two factors.



$S^+ = \{X_i \text{ "above" } X_k\}$     $S^- = \{X_i \text{ below } X_k\}$

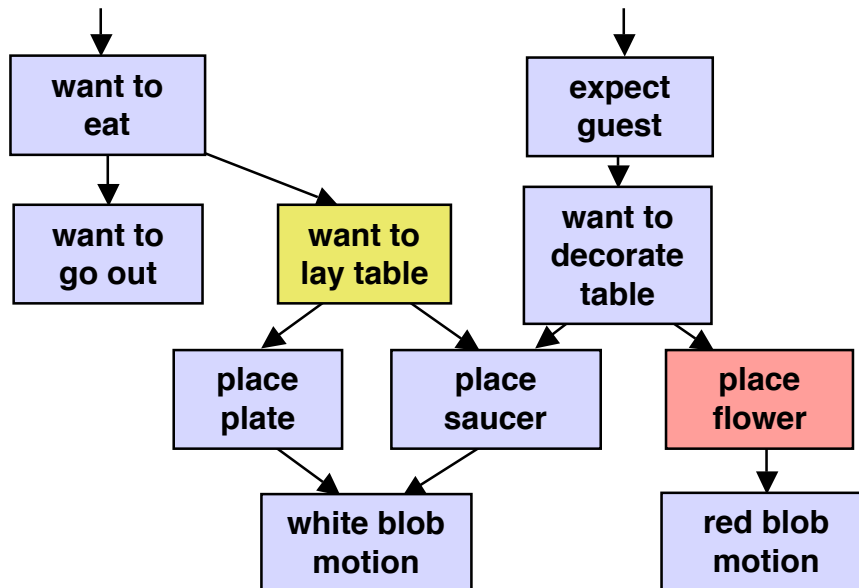
$$\begin{aligned}
 P(X_k=c) &= \sum_{X_i \neq X_k} P(X_1 \dots X_k=c \dots X_N) \\
 &= \sum_{X_i \neq X_k} P(X_k=c | \text{Pa}(X_k)) \prod_{X_i \neq X_k} P(X_i | \text{Pa}(X_i)) \\
 &= \left[ \sum_{X_i \in S^+} P(X_k=c | \text{Pa}(X_k)) \prod_{X_i \in S^+} P(X_i | \text{Pa}(X_i)) \right] \cdot \left[ \sum_{X_i \in S^-} \prod_{X_i \in S^-} P(X_i | \text{Pa}(X_i)) \right] \\
 &= \pi(X_k=c) \cdot \lambda(X_k=c) \quad \Rightarrow \text{propagation scheme is possible}
 \end{aligned}$$

# Approximate Inference in Bayesian Networks

- Inference in singly-connected Bayes Nets can be computed with  $O(N)$
- Worst-case complexity in general Bayes Nets is exponential, hence approximate algorithms with less complexity are useful.

## Basic idea:

Use random sampling (Monte Carlo method) to compute the approximate probability of an event based on a JPD and evidence.



Example: Determine  $P(\text{"place flower"} \mid \text{"want to lay table"})$

- Draw sample for each node based on probability conditioned on parent samples
- Repeat process many times
- Relative frequency of samples matching evidence converges to correct result in the limit.

# Sampling Methods

## Direct Sampling:

Estimate the probability of an event without evidence by sampling a Bayes Net.

## Recommended Reading:

Russell & Norvig:  
Artificial Intelligence - A  
Modern Approach, 2nd  
Ed., Prentice Hall, 2003

## Rejection Sampling:

Estimate the probability of an event by sampling a Bayes Net and discarding all samples which do not match the evidence

## Sampling with Likelihood Weighting:

Estimate the probability of an event by sampling a Bayes Net and weighting all samples according to their likelihood to generate the evidence

All three methods generate consistent estimates (which converge to the true value).

# Hidden Markov Models

A sequence of observations may be governed by underlying probabilistic state transitions.

Example: A person laying a table may plan to first place the plates, then the cups, then the cutlery in a cyclic order (with a chance to deviate from this order).

As usual in vision, observations may be disturbed and may provide uncertain evidence about the current state.

Such phenomena may be modelled by a Hidden Markov Model (HMM).

A (discrete) HMM is defined by

- a finite number of states  $a_1, a_2, \dots, a_K$
- a sequence of state transition events  $t_0, t_1, \dots, t_n$  (not necessarily times)
- probabilities of state transitions  $p_{ij}$  from state  $i$  to state  $j$ , each depending only on the previous state
- observations  $b_1, b_2, \dots, b_M$  probabilistically related to each state
- probabilities  $q_{km}$  which map states into observations

## Notation for HMM

- sequence of random variables  $X^{(1)}, \dots, X^{(n)}$  (state variables) with values from  $\{a_1, \dots, a_K\}$
- Markov Chain property of  $X^{(1)}, \dots, X^{(n)}$ :  $P(X^{(n)}|X^{(n-1)} \dots X^{(1)}) = P(X^{(n)}|X^{(n-1)})$ 
  - if  $P(X^{(n)}|X^{(n-1)})$  is independent of  $n$ , the Markov Chain is homogeneous
  - transition probabilities  $P(X^{(n)}=a_i|X^{(n-1)}=a_j)$  are represented by the state transition matrix

$$W^{(n)} = \begin{bmatrix} p_{11} & \dots & p_{1K} \\ \vdots & & \\ p_{K1} & \dots & p_{KK} \end{bmatrix}$$

- random variables  $Y^{(1)}, \dots, Y^{(n)}$  (observations) with values from  $\{b_1, \dots, b_M\}$
- observation probabilities  $P(Y^{(n)}|X^{(n)})$  are represented by the matrix

$$Q = \begin{bmatrix} q_{11} & \dots & q_{1M} \\ \vdots & & \\ q_{K1} & \dots & q_{KM} \end{bmatrix}$$

- initial probabilities  $\underline{\pi}^T = [ P(X^{(1)}=a_1) \ P(X^{(1)}=a_2) \ \dots \ P(X^{(1)}=a_K) ]$

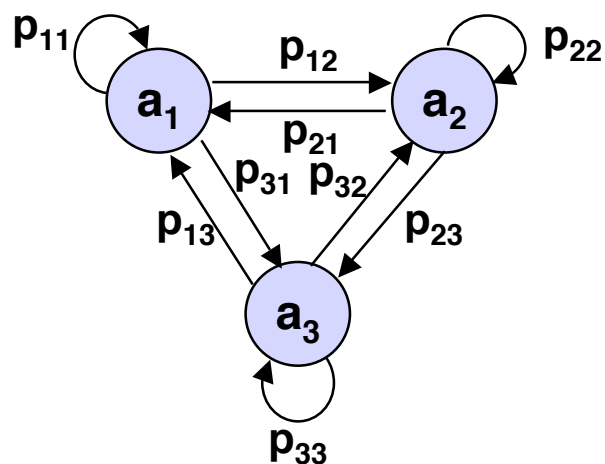
# Properties of a Homogeneous HMM

Probability vector for state  $X^{(2)}$ :  $\underline{\pi}^{(2)} = W^T \underline{\pi}$

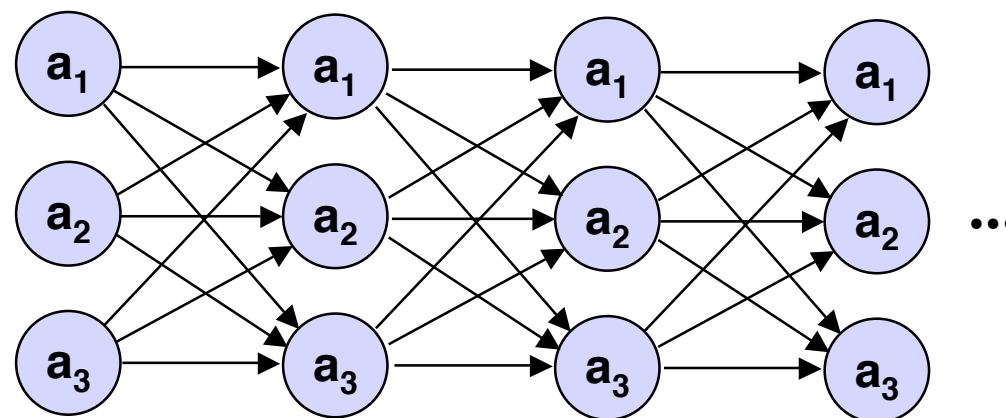
Probability vector for state  $X^{(n)}$ :  $\underline{\pi}^{(n)} = (W^T)^{n-1} \underline{\pi}$

There is always a stationary distribution  $\underline{\pi}_s$  such that  $\underline{\pi}_s = W^T \underline{\pi}_s$

Graphical representation:



Trellis ("Spalier") representation:



- each (directed) path corresponds to a legal sequence of states
- the probability of a path is equal to the product of the transition probabilities



# Paths through a HMM

Given a sequence of  $N$  observations, we want to find the most probable sequence of states which may have led to the observations.

## Extension of trellis representation

- arc weights leading into states  $X^{(n)}$ : transition probabilities  $p_{ij}$
- node weights of states  $X^{(n)}$ : observation likelihoods  $q_{jm}$  for given observations  $Y^{(n)} = b_{m_n}$
- product of initial probability and node and arc probabilities along path:  $P(Y^{(1)}=b_{m_1}, \dots, Y^{(N)}=b_{m_N}, X^{(1)}=a_{k_1}, \dots, X^{(N)}=a_{k_N})$   
probability of observations and states

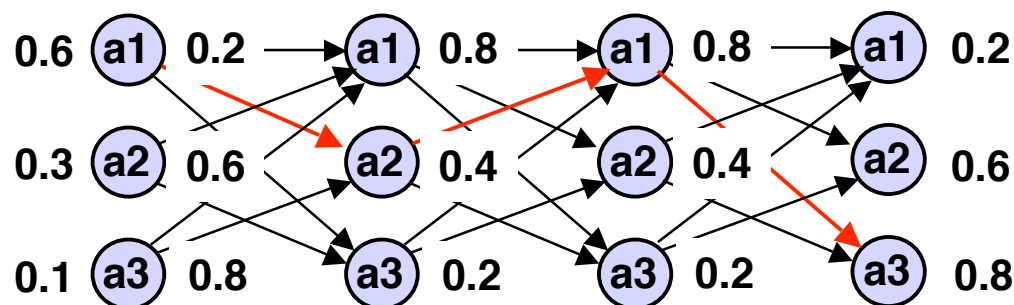
### Example:

$$W = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.4 & 0.6 & 0.0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix}$$

observations  
 $b_2, b_1, b_1, b_2$



probability of observations along path are

$$P(Y^{(1)}=b_2, Y^{(2)}=b_1, Y^{(3)}=b_1, Y^{(4)}=b_2, \text{ states of path}) =$$

$$0.6 \cdot 0.2 \cdot 0.2 \cdot 0.4 \cdot 0.1 \cdot 0.8 \cdot 0.5 \cdot 0.8$$

# Finding Most Probable Paths

The most probable sequence of states is found by maximizing

$$\max_{k_1 \dots k_N} P(X^{(1)}=a_{k_1}, \dots, X^{(N)}=a_{k_N} \mid Y^{(1)}=b_{m_1}, \dots, Y^{(N)}=b_{m_N}) = \max_{\underline{a}} P(\underline{a} \mid \underline{b})$$

Equivalently, the most probable sequence of states follows from

$$\max_{\underline{a}} P(\underline{a} \mid \underline{b}) = \max_{\underline{a}} P(\underline{a} \mid \underline{b}) P(\underline{b})$$

Hence the maximizing sequence of states can be found by exhaustive search of all path probabilities in the trellis. However, complexity is  $O(K^N)$  with  $K$  = number of different states and  $N$  = length of sequence.

The Viterbi Algorithm does the job in  $O(KN)$ !

Overall maximization may be decomposed into a backward sequence of maximizations:

$$\begin{aligned} \max_{\underline{a}} P(\underline{a} \mid \underline{b}) &= \max_{k_1 \dots k_N} \pi_{k_1} q_{k_1 m_1} \prod_{n=2 \dots N} p_{k_{n-1} k_n} q_{k_{n-1} m_n} \\ &= \max_{k_1} \pi_{k_1} q_{k_1 m_1} \left( \max_{k_2} p_{k_1 k_2} q_{k_2 m_2} \left( \dots \left( \max_{k_N} p_{k_{N-1} k_N} q_{k_{N-1} m_N} \right) \dots \right) \right) \end{aligned}$$

**Step N**

**Step N-1**

**Step 1**

# Example for Viterbi Algorithm

Typical maximization step of Viterbi algorithm:

$$\max_{k_n} \{ p_{k_{n-1}k_n} \cdot q_{k_{n-1}m_n} \cdot \langle \text{result of previous maximization step} \rangle \}$$

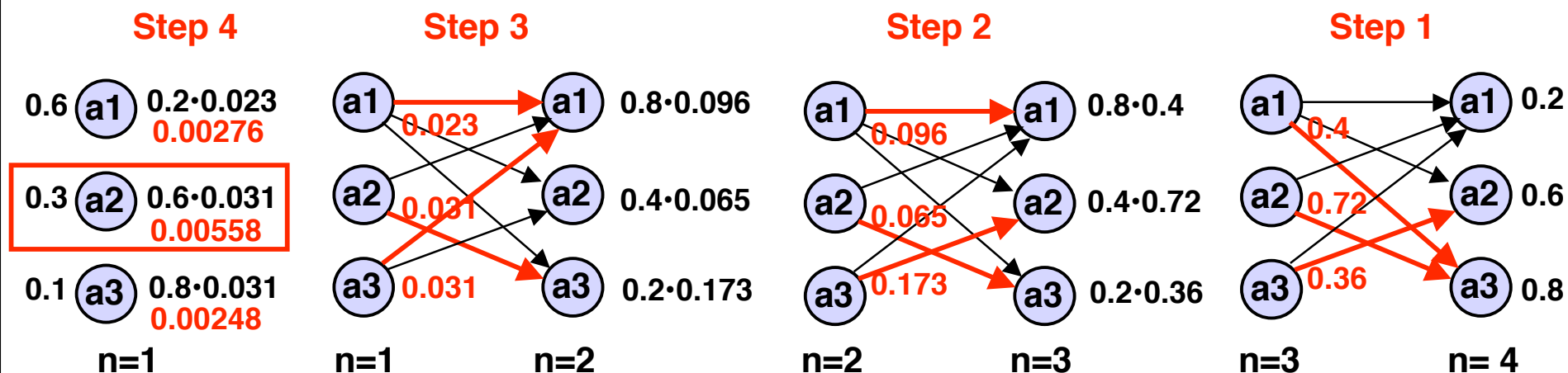
Example as earlier:

$$W = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.4 & 0.6 & 0.0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix}$$

observations  
b<sub>2</sub>, b<sub>1</sub>, b<sub>1</sub>, b<sub>2</sub>



red numbers show maximization results, red arrows maximizing transitions

# Model Evaluation for Given Observations

What is the likelihood that a particular HMM (out of several possible models) has generated the observations?

Likelihood of observations given model:

$$P(Y^{(1)}=b_{m_1}, \dots, Y^{(N)}=b_{m_N} \mid \text{model}) = P(\underline{b}) = \sum_{\underline{a}} P(\underline{a} \mid \underline{b})$$

Instead of summing over all  $\underline{a}$ , one can use a forward algorithm based on the recursive formula:

$$\begin{aligned} & \underline{P(a_j^{(n+1)}, b_{m_1}, \dots, b_{m_n}, b_{m_{n+1}})} \\ &= P(a_j^{(n+1)}, b_{m_1}, \dots, b_{m_n}) \cdot P(b_{m_{n+1}} \mid a_j^{(n+1)}) \\ &= \sum_i [P(a_j^{(n+1)}, a_i^{(n)}, b_{m_1}, \dots, b_{m_n})] \cdot P(b_{m_{n+1}} \mid a_j^{(n+1)}) \\ &= \sum_i [P(a_j^{(n+1)} \mid a_i^{(n)}, b_{m_1}, \dots, b_{m_n}) P(a_i^{(n)}, b_{m_1}, \dots, b_{m_n})] \cdot P(b_{m_{n+1}} \mid a_j^{(n+1)}) \\ &= \sum_i [P(a_j^{(n+1)} \mid a_i^{(n)}) \cdot P(a_i^{(n)}, b_{m_1}, \dots, b_{m_n})] \cdot P(b_{m_{n+1}} \mid a_j^{(n+1)}) \\ &= \sum_i [p_{ij} \cdot \underline{P(a_i^{(n)}, b_{m_1}, \dots, b_{m_n})}] \cdot q_{j m_{n+1}} \end{aligned}$$

$$\text{Finally: } P(b_{m_1}, \dots, b_{m_N}) = \sum_i P(a_i^{(n+1)}, b_{m_1}, \dots, b_{m_N})$$

# Example for Model Evaluation (1)

Computing the probability of observations stepwise as they come in.

Example as earlier:

$$W = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.4 & 0.6 & 0.0 \end{bmatrix} \quad Q = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix} \quad \pi = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix} \quad \text{observations} \\ \mathbf{b}_2, \mathbf{b}_1, \mathbf{b}_1, \mathbf{b}_2$$

## Step 1

$$P(\mathbf{a}_j^{(1)}, \mathbf{b}_{m_1}) = \pi_j \cdot q_{j m_1}$$

$$P(\mathbf{a}_1^{(1)}, \mathbf{b}_2) = 0.6 \cdot 0.2 = 0.12$$

$$P(\mathbf{a}_2^{(1)}, \mathbf{b}_2) = 0.3 \cdot 0.6 = 0.18$$

$$P(\mathbf{a}_3^{(1)}, \mathbf{b}_2) = 0.1 \cdot 0.8 = 0.08$$

Note that  $P(\mathbf{b}_{m_1}, \dots, \mathbf{b}_{m_n})$  can be computed after each step by summing out the dependency on the state  $X^{(n)}$ .

## Step 2

$$P(\mathbf{a}_j^{(2)}, \mathbf{b}_{m_1}, \mathbf{b}_{m_2}) = \sum [ p_{ij} \cdot P(\mathbf{a}_i^{(1)}, \mathbf{b}_{m_1}) ] \cdot q_{j m_2}$$

$$P(\mathbf{a}_1^{(2)}, \mathbf{b}_2, \mathbf{b}_1) = [ 0.3 \cdot 0.12 + 0.1 \cdot 0.18 + 0.4 \cdot 0.08 ] \cdot 0.8 = 0.0314$$

$$P(\mathbf{a}_2^{(2)}, \mathbf{b}_2, \mathbf{b}_1) = [ 0.2 \cdot 0.12 + 0.6 \cdot 0.08 ] \cdot 0.4 = 0.0288$$

$$P(\mathbf{a}_3^{(2)}, \mathbf{b}_2, \mathbf{b}_1) = [ 0.5 \cdot 0.12 + 0.9 \cdot 0.18 ] \cdot 0.2 = 0.0072$$

## Example for Model Evaluation (2)

Example continued:

$$W = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.4 & 0.6 & 0.0 \end{bmatrix} \quad Q = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix} \quad \pi = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix} \quad \text{observations} \\ b_2, b_1, b_1, b_2$$

### Step 3

$$P(a_j^{(3)}, b_{m_1}, b_{m_2}, b_{m_3}) = \sum [ p_{ij} \cdot P(a_j^{(2)}, b_{m_1}, b_{m_2}) ] \cdot q_{j m_3}$$

$$P(a_1^{(3)}, b_2, b_1, b_1) = [ 0.3 \cdot 0.0314 + 0.1 \cdot 0.0288 + 0.4 \cdot 0.0072 ] \cdot 0.8 = 0.01214$$

$$P(a_2^{(3)}, b_2, b_1, b_1) = [ 0.2 \cdot 0.0314 + 0.6 \cdot 0.0072 ] \cdot 0.4 = 0.00424$$

$$P(a_3^{(3)}, b_2, b_1, b_1) = [ 0.5 \cdot 0.0314 + 0.9 \cdot 0.0288 ] \cdot 0.2 = 0.00832$$

### Step 4

$$P(a_j^{(4)}, b_{m_1}, b_{m_2}, b_{m_3}, b_{m_4}) = \sum [ p_{ij} \cdot P(a_j^{(3)}, b_{m_1}, b_{m_2}, b_{m_3}) ] \cdot q_{j m_4}$$

$$P(a_1^{(4)}, b_2, b_1, b_1, b_2) = [ 0.3 \cdot 0.01214 + 0.1 \cdot 0.00424 + 0.4 \cdot 0.00832 ] \cdot 0.2 = 0.001479$$

$$P(a_2^{(4)}, b_2, b_1, b_1, b_2) = [ 0.2 \cdot 0.01214 + 0.6 \cdot 0.00832 ] \cdot 0.6 = 0.004452$$

$$P(a_3^{(4)}, b_2, b_1, b_1, b_2) = [ 0.5 \cdot 0.01214 + 0.9 \cdot 0.00424 ] \cdot 0.4 = 0.003954$$

### Final step

$$P(b_{m_1}, b_{m_2}, b_{m_3}, b_{m_4}) = \sum P(a_j^{(4)}, b_{m_1}, b_{m_2}, b_{m_3}, b_{m_4}) = \boxed{0.009885}$$