Uncertainty

Professor Marie Roch

12.2-12.6,
13-13.3.3.1
(except 13.2.1),
Rabiner’s HMM Tutorial

Uncertainty

$$toothache \Rightarrow cavity$$

What else can cause a toothache?

$$toothache \Rightarrow cavity \lor cracked\ tooth \lor stuck\ popcorn \lor \ldots$$

Logic can fail us:

- laziness – Too difficult to enumerate rules without exceptions
- theoretical ignorance – May not fully understand the system
- practical ignorance – System may not be fully observable

An agent’s view

Probabilities represent a level of belief in a world

Decision-theoretic agents

function DT-AGENT(*percept*) **returns** an *action*
  persistent: *belief_state*, probabilistic beliefs about the current state of the world
    action, the agent's action

  update *belief_state* based on *action* and *percept*
  calculate outcome probabilities for actions,
    given action descriptions and current *belief_state*
  select *action* with highest expected utility
    given probabilities of outcomes and utility information
  return *action*

Basic probability

- Random variables represent an outcome or world, e.g. X represents a die roll, $W_{2,1}$ is the Wumpus in cave 2,1
- P(X) is the probability of X happening
- It is common to use
  - CAPITALS to represent outcomes in general: P(X)
  - lower case to denote specific outcomes: P(x=5)
- Probability distributions characterize probability over all outcomes and require:

$$\forall x \in \mathrm{domain}(X), 0 \leq P(X = x) \leq 1$$

$$\sum_{x \in \mathrm{domain}(X)} P(X = x) = 1$$

Posterior (conditional) probability

- Posterior probability is conditioned on another event

  What is the probability that I have a cavity *given* that I have a toothache?

  $$P(cavity|toothache)$$

- In contrast, prior probabilities have no condition

  $$P(cavity)$$

- Definition: $\mathrm{P(A|B)} = \frac{P(A \wedge B)}{P(B)}$ or equivalently: $P(A \wedge B)$=P(A|B)P(B)

  product rule

note: $P(A \wedge B)$ is frequently written as $P(A, B)$

Propositions

- Let us consider random variable values as possible worlds (like our model checking in propositional logic)
- If we want to know P of proposition ϕ holding:

$$P(\phi) = \sum_{\omega \in \phi} P(\omega)$$

- In addition

$$P(\neg\phi) = \sum_{\omega \in \neg\phi} P(\omega) = 1 - \sum_{\omega \in \phi} P(\omega)$$

and

$$P(\phi \vee \rho) = P(\phi) + P(\rho) - P(\phi \wedge \rho)$$

inclusion-exclusion principle

Joint probabilities

- Probability of multiple things, e.g. $P(A, B, C)$.
- Can be decomposed with the product rule:

$$\begin{aligned} P(A, B, C) &= P((A, B), C) \\ &= P(C|A, B)P(A, B) \\ &= P(C|A, B)P(B|A)P(A) \end{aligned}$$

  This is called the chain rule.

  In general: $P(x_1, \dots, x_n) = \Pi_{i=1}^{n} P(x_i | x_{i-1}, x_{i-2}, \dots, x_1)$
- If A and B are independent, then $P(B|A) = P(B)$
- In general, joint probabilities of independent variables can be multiplied: $P(A, B, C) = P(A)P(B)P(C)$ (A/B/C independent)

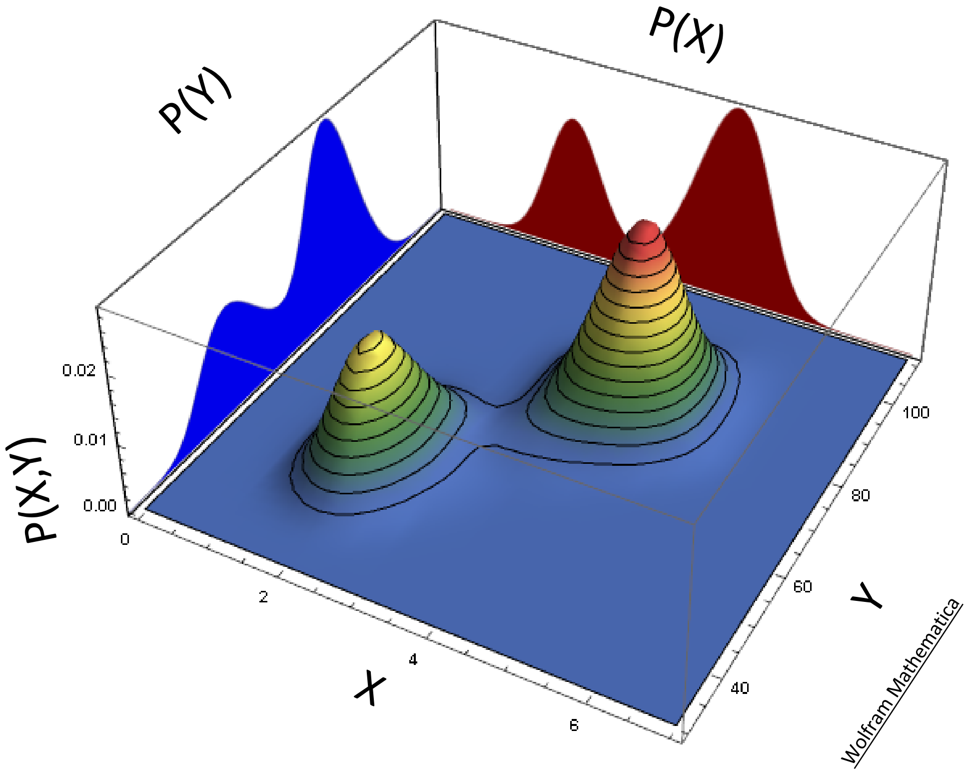
Marginalization

- Suppose we know the joint probability between X and Y, P(X,Y), and want P(X)

$$P(X) = \sum_{y} P(X, Y = y)$$

example:

$$P(Eat, Rent) = \sum_{r=0}^{owed} P(Eat, Rent = r)$$



Bayes' rule

(of conditional probability)

Remember definition posterior probability $P(A|B) = \frac{P(A,B)}{P(B)}$

$$\mathrm{P(B|A)} = \frac{P(B,A)}{P(A)} \rightarrow\ P(B,A) = P(B|A)P(A)$$

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B,A)}{P(B)}$$

$$\therefore\ P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

also known as Bayes' Theorem/Bayes' Law)

Bayes' rule: Why do we care?

Suppose we observe an effect.

- Knowing the cause can be difficult
- Simpler to estimate P(effect | cause); Bayes' rule lets us turn this around:

$$P(Disease|Symptom) = \frac{P(Symptom|Disease)P(Disease)}{P(Symptom)}$$

If we are looking at multiple diseases, we do not need P(Symptom) to make a choice between them.

We can treat ${}^{1}/_{P(Symptom)}$ as a constant α:

$$P(Disease|Symptom) = \alpha P(Symptom|Disease)P(Disease)$$

Bayes' rule example

- A symptom of meningitis is a stiff neck

$$P(s|m) = 0.7$$

but the case rate for stiff necks is low and meningitis very low

$$P(s) = 0.01, P(m) = 1/50000$$

$$P(m|s) = \frac{P(S|m)P(m)}{P(s)} = \frac{0.7 \times {}^{1}/_{50000}}{0.01} = 0.0014$$

Conditional independence

What do you see?
What does your neighbor see?

Conditional independence

- Both Shyam and Monica observe the cloud
- If they haven't talked to each other about what they saw, the probability conditioned on a specific cloud is independent

$$P(S, M|c) = P(S|c)P(M|c)$$

Naïve Bayes models

- Exploit conditional independence to make simple models
- If a cause has *n* effects that are conditionally independent:

$$P(cause, e_1, e_2, \ldots, e_n) = P(cause) \prod_{i=1}^{n} P(e_i | cause)$$

- Naïve Bayes models imply that we don't really know if our effects are conditionally independent, but we assume so anyway.
- If we wanted to find $P(cause | e_1, e_2, \ldots, e_n)$ we could use the product rule and conditional independence:

$$P(cause | e_1, e_2, \ldots, e_n) = \alpha P(cause) \prod_{i=1}^{n} P(e_i | cause)$$

Example: Sentence to category

Disneyland raised its entrance price by thirty percent.

We might ask the question: Is this about business or entertainment?

We could consider how often articles are about each of these categories (prior):

$$P(\text{business}) = .03$$

$$P(\text{entertainment}) = .04$$

Example: Sentence to category

Bayes factors:

P(Disneyland|business) = .2

P(Disneyland|entertainment) = .8

P(price|business) = .9

P(price|entertainment) = .1

Prior probabilities:

P(business) = .03

P(entertainment) = .04

$$P(bz|Dland, \$)$$
$$= P(Dland|bz)P(\$|bz)$$
$$= .03 \cdot .2 \cdot .9 = .0054$$

$$P(ent|Dland, \$)$$
$$= P(ent)P(Dland|ent)P(\$|ent)$$
$$= \ .04 \cdot .8 \cdot .1 = .0032$$

We classify as the sentence as the category that maximizes P

Probabilistic reasoning

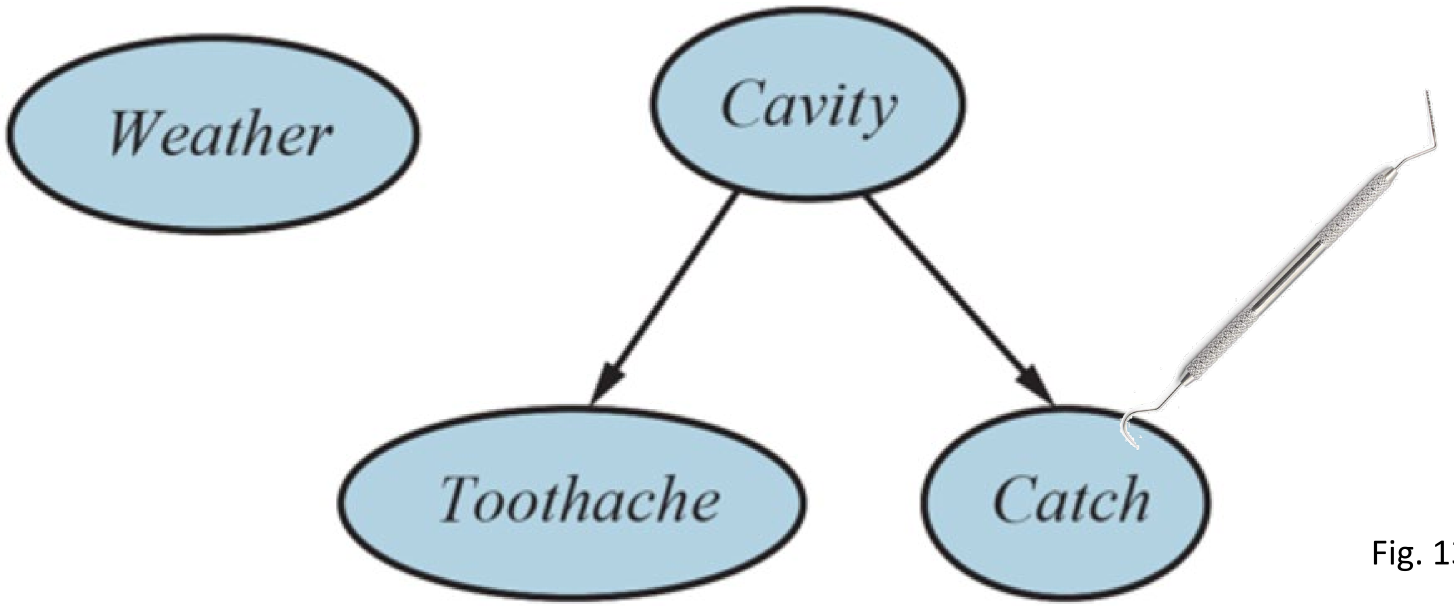


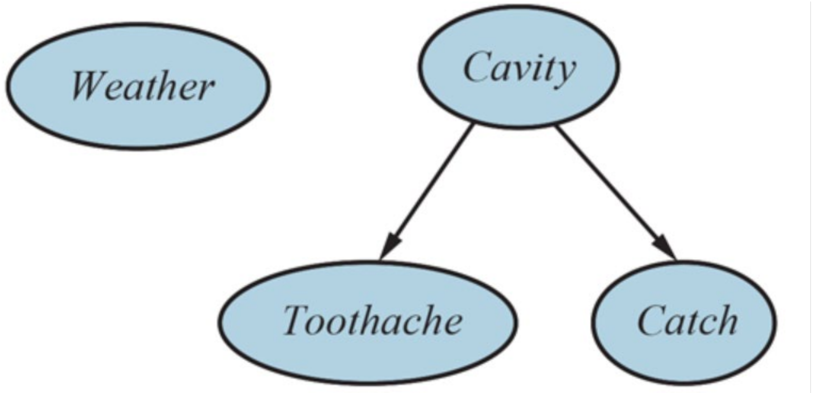
Fig. 13.1

Bayesian network

- Having a cavity influences the likelihood of a toothache or a dentist's sickle probe to catch on your tooth
- Changes in weather do not cause toothaches or probe catches.

Bayesian networks

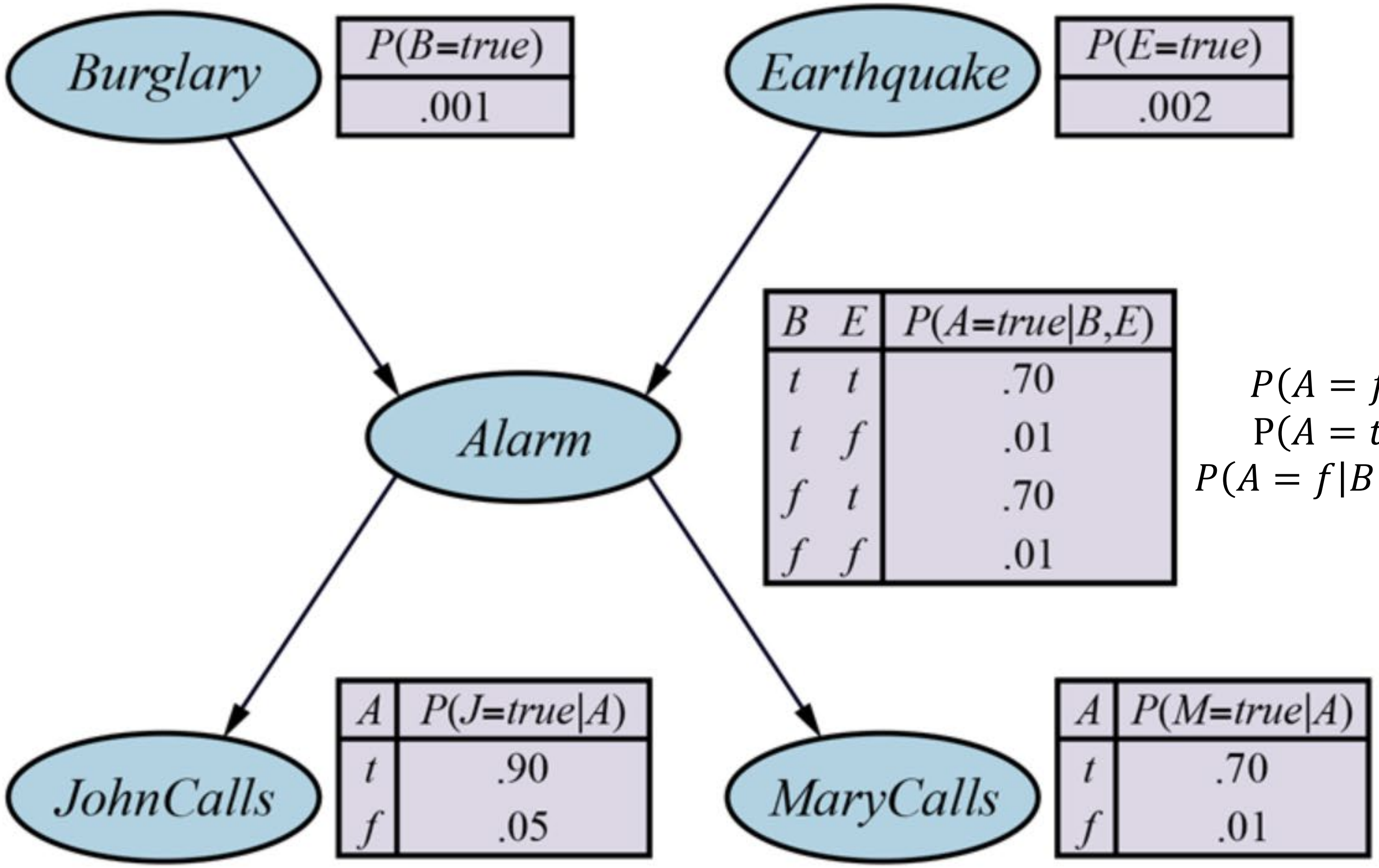
- Nodes are random variables
- Variables can be connected by directed arcs that do not form cycles
- Each variable *V* has
  - prior probability (no parents): P(V)
  - conditional probability P(V|parents(V))
- Forms a directed acyclic graph



Pearl's Bayesian network example

- Burglar alarm set off by
  - Burglar
  - Earthquake
- Neighbors Mary and John have agreed to let you know when they hear the alarm
  - Mary listens to headphones, and often misses the alarm
  - Your home telephone ringtone is similar to the alarm (silly you) and John sometimes calls you when your phone rings
    (yes, you still have a landline)

Pearl's Bayesian network example



P(B=true)
.001

P(E=true)
.002

B	E	P(A=true|B,E)
t	t	.70
t	f	.01
f	t	.70
f	f	.01

A	P(J=true|A)
t	.90
f	.05

A	P(M=true|A)
t	.70
f	.01

$P(A = f|conditions)$ implicit
$\mathrm{P}(A = t|B = t, E = t) = .7 \rightarrow$
$P(A = f|B = t, E = t) = 1 - .7 = .3$

Fig. 13.2 R&N

Bayes net semantics

- $P(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} P(x_i | parents(X_i))$

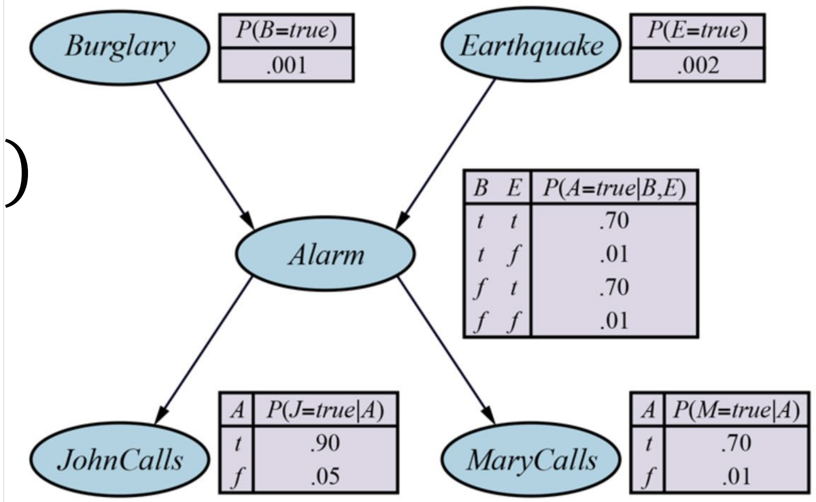


Fig. 13.2 R&N

- Consider: alarm sounds with neither burglary/earthquake and both neighbors call

$$P(j, m, a, \neg b, \neg e) = P(j|a)P(m|a)P(a|\neg b \wedge \neg e)P(\neg b)P(\neg e)$$
$$= .9 \times .7 \times .01 \times .999 \times .998 = .00628$$

Bayesian networks are frequently called Bayes nets

Bayes net semantics

- We can compute the marginal to answer just about any question related to this, e.g. John & Mary call when the alarm sounds and there is no burglary

$$P(j, m, a, \neg b) = \sum_{E \in e, \neg e} P(j|a)P(m|a)P(a|\neg b \wedge E)P(\neg b)P(E)$$

- Note that earthquake was not specified in the question;
  we computed the marginal probability to integrate/sum it out.

Constructing a Bayes network

- Nodes
  - Determine required random variables
  - Number them $X_1, X_2, \ldots, X_n$ (better if causes precede effects)
- Network edges

  for i = 1:n

  Find minimum parents(X_i) such that $P(X_i | X_{i-1}, \ldots, X_1) = P(X_i | parents(X_i))$

  Add edges $parents(X_i)$ to X_i

  Estimate conditional probability table $P(X_i | parents(X_i))$

Note: We are only concerned about direct influence, so Alarm influences MaryCalls, but Burglary and Earthquake do not.

Estimating the conditional probability table

- Estimated from training data
- Discrete – Use a frequentist model, e.g., $P(A = a|b) = \frac{count(a,b)}{count(b)}$
- Continuous - Fit distribution to data, e.g. $P(A = a|b) \sim n(\mu, \sigma^2)$, use the mean and variance of examples where B=b

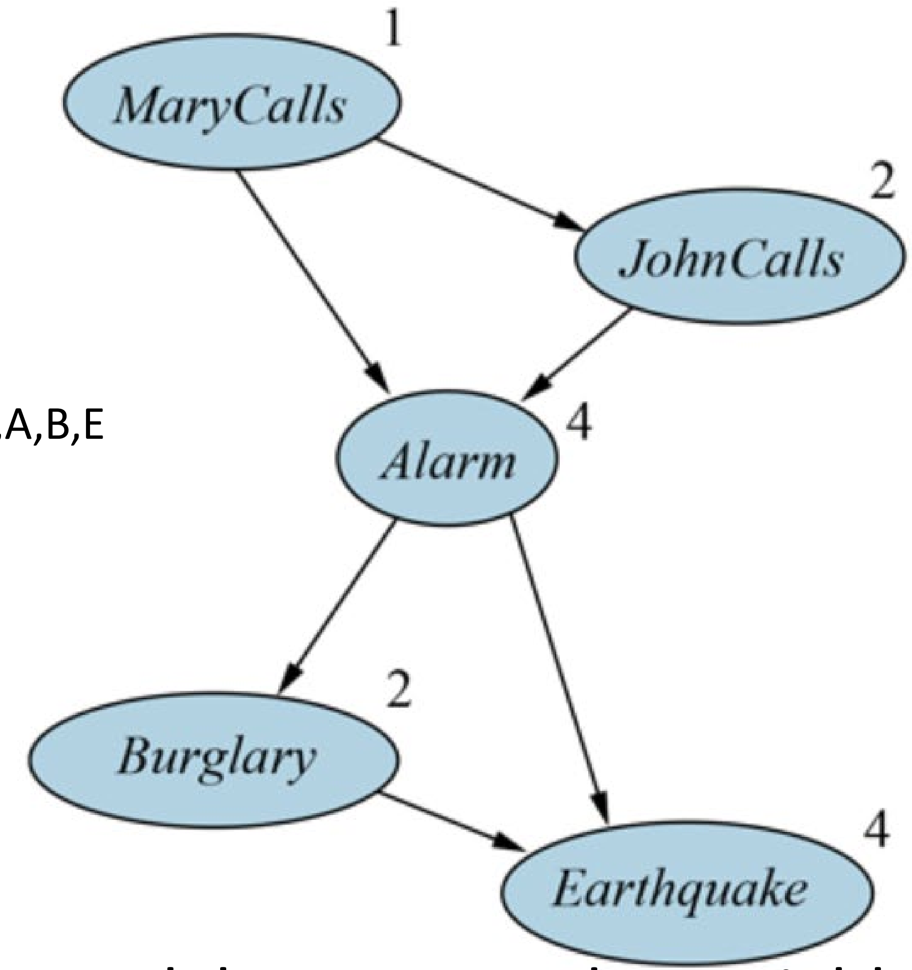
- Chapter 20 has more details on learning in probabilistic models. (See also any basic statistics book's chapter on maximum likelihood estimation)

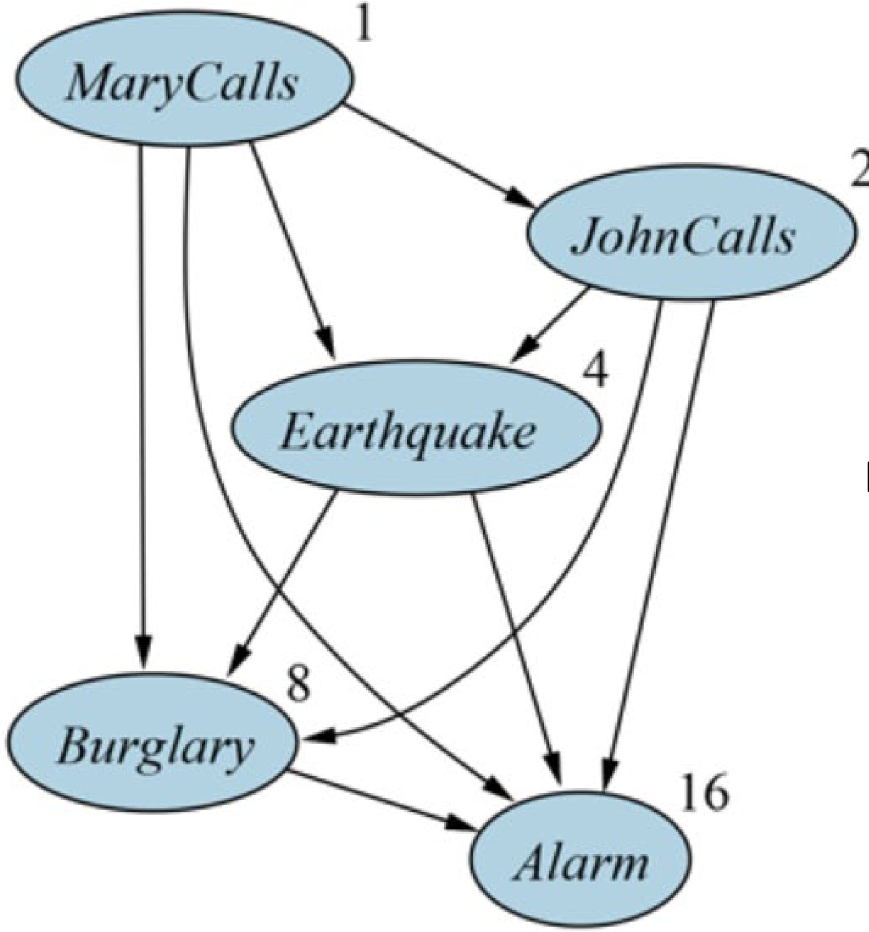
Variable order in Bayes network construction

- Construction depends on order
- Consider order: MaryCalls, JohnCalls, Alarm, Burglary, Earthquake
  - MC – no parents
  - JC – If MC, probably an earthquake, hence P(JC|MC)
  - A – Alarm more likely if both MC and JC call, therefore they are parents of A.
  - B – If we know alarm state, MC & JC do not give us additional information about whether this was a burglary or earthquake, hence P(B|A)
    [We assume that this is a minor earthquake, an example of laziness in uncertainty.]
  - E – If A, then it is more likely that there was an earthquake, but B would also cause an alarm and knowing this reduces the probability of E. Hence P(E|A,B)

Order of variables in construction matters

M,J,A,B,E





M,J,E,B,A

In general, better to order variables in what we think might be a causal manner. However, both networks will learn appropriate distributions.

Efficient representations

- For a binary Bayes net with at most k parents, conditional probability tables (CPT) have $O\left(2^k\right)$ entries.
- Many times, relationships fit into patterns that we call **canonical distributions**, and can specify the conditional probability tables with the canonical name and a few parameters.

skip 13.2.1

Canonical distribution examples

- deterministic nodes – are not probabilistic but can be represented by a function, e.g. *ReservoirLevelChange* might be the sum of inputs from rivers – evaporation
- context-specific independence – Parents might be independent when other parents have specific values

Example: `P(Damage|Ruggedness,Accident) = d₁ if Accident == false`
`else d₂(Ruggedness)`

$$P_{d_1}(Damage) = \begin{cases} .995 & Damage = false \\ .005 & Damage = true \end{cases}$$

indep. of ruggedness

$$P_{d_2}(Damage|Accident) = \begin{cases} f_{Ruggedness}(.20) & Damange = false \\ f_{Ruggedness}(.80) & Damage = true \end{cases}$$

Canonical distribution examples

- noisy-or – Permits uncertainty in causation
  e.g., In propositional logic we might state: $Fever \iff Cold \lor Flu \lor Malaria$
  If you have one of these, you have a fever.

- Suppose disease *i* occurs without fever with frequency q_i :

$$q_{cold} = P(\neg fever | cold, \neg flu, \neg malaria) = 0.6$$
$$q_{flu} = P(\neg fever | \neg cold, flu, \neg malaria) = 0.2$$
$$q_{malaria} = P(\neg fever | \neg cold, \neg flu, malaria) = 0.1$$

Noisy-or would make fever true as follows:

$$P(fever | parents(fever)) = 1 - \prod_{\substack{j: j = true \land \\ j \in parents(fever)}} q_j$$

We can think of this as 1 – the joint probability that everything you have that is making you sick did not cause a fever.

Noisy-or example

$$q_{\text{cold}} = P(\neg fever|cold, \neg flu, \neg malaria) = 0.6,$$
$$q_{\text{flu}} = P(\neg fever|\neg cold, flu, \neg malaria) = 0.2,$$
$$q_{\text{malaria}} = P(\neg fever|\neg cold, \neg flu, malaria) = 0.1.$$

Fig. 13.5

Cold	*Flu*	*Malaria*	$P(fever \mid \cdot)$	$P(\neg fever \mid \cdot)$
f	*f*	*f*	0.0	1.0
f	*f*	*t*	0.9	**0.1**
f	*t*	*f*	0.8	**0.2**
f	*t*	*t*	0.98	$0.02 = 0.2 \times 0.1$
t	*f*	*f*	0.4	**0.6**
t	*f*	*t*	0.94	$0.06 = 0.6 \times 0.1$
t	*t*	*f*	0.88	$0.12 = 0.6 \times 0.2$
t	*t*	*t*	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Bayesian nets with continuous variables

- Several options
  - Discretize – split into discrete values based on range
  - Use a parametric distribution, e.g., normal distribution
  - Non-parametric options possible, but beyond our scope

- Linear-Gaussian conditional distribution
  - Most common parametric distribution
  - Variance fixed, mean dependent on a continuous parent

Hybrid Bayesian nets

- Contain both discrete and continuous variables

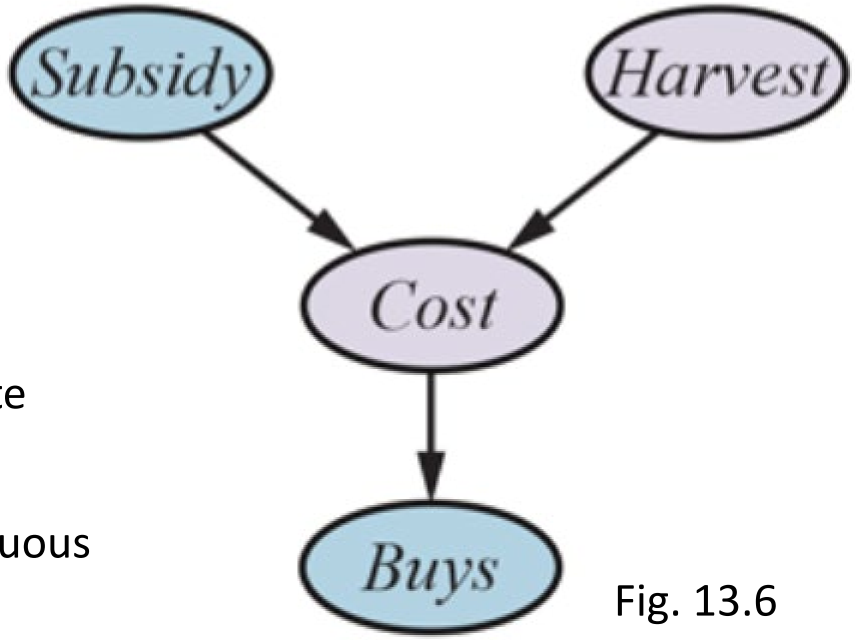
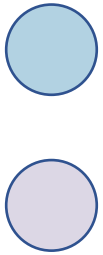


Fig. 13.6



Discrete

Continuous

Whether or not a consumer purchases fruit depends on its cost.

The cost depends on the harvest and whether or not a government subsidy was provided.

Linear-Gaussian example

$$P(c|h, subsidy) = N(c; a_t h + b_t, \sigma_t^2) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_t h + b_t)}{\sigma_t}\right)^2}$$

$$P(c|h, \neg subsidy) = N(c; a_f h + b_f, \sigma_f^2) = \frac{1}{\sigma_f\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c-(a_f h + b_f)}{\sigma_f}\right)^2}$$

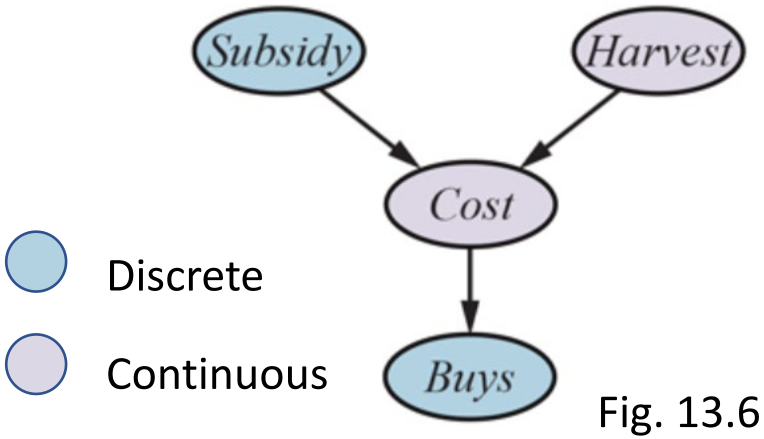


Fig. 13.6

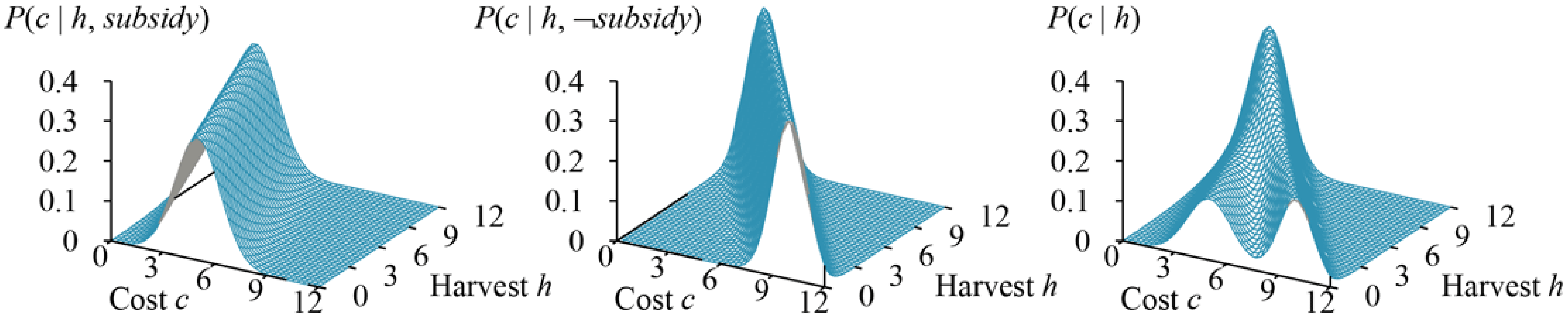


Fig. 13.7

Discrete with continuous parents

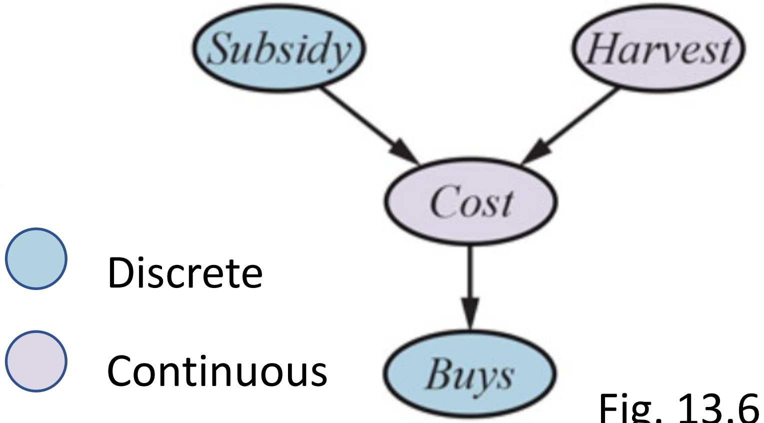
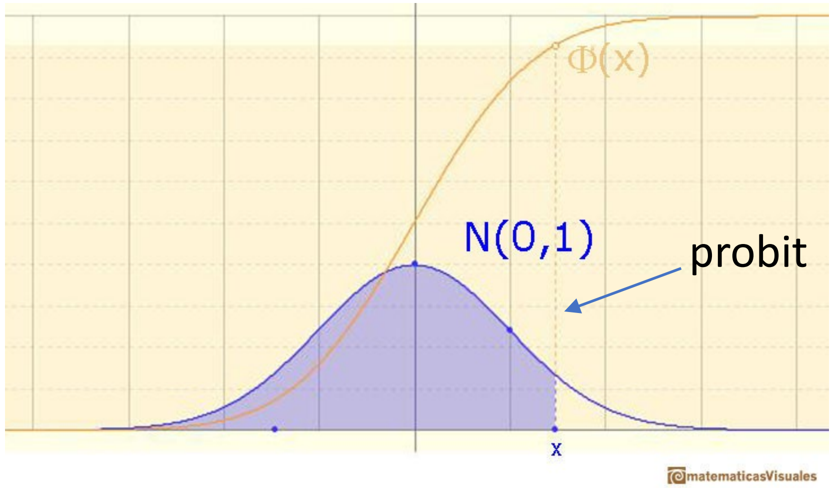


Fig. 13.6

- We need some type of "soft" threshold
- Remember the cumulative density function we introduced with the χ^2 distribution. In general, $P(X \leq x)$ (sometimes denoted $\Phi(x)$):

$$P(X \leq x) = \int_{-\infty}^{x} P(x)dx$$

- However, this varies smoothly from 0 to 1 as X increases, which is not exactly what we want

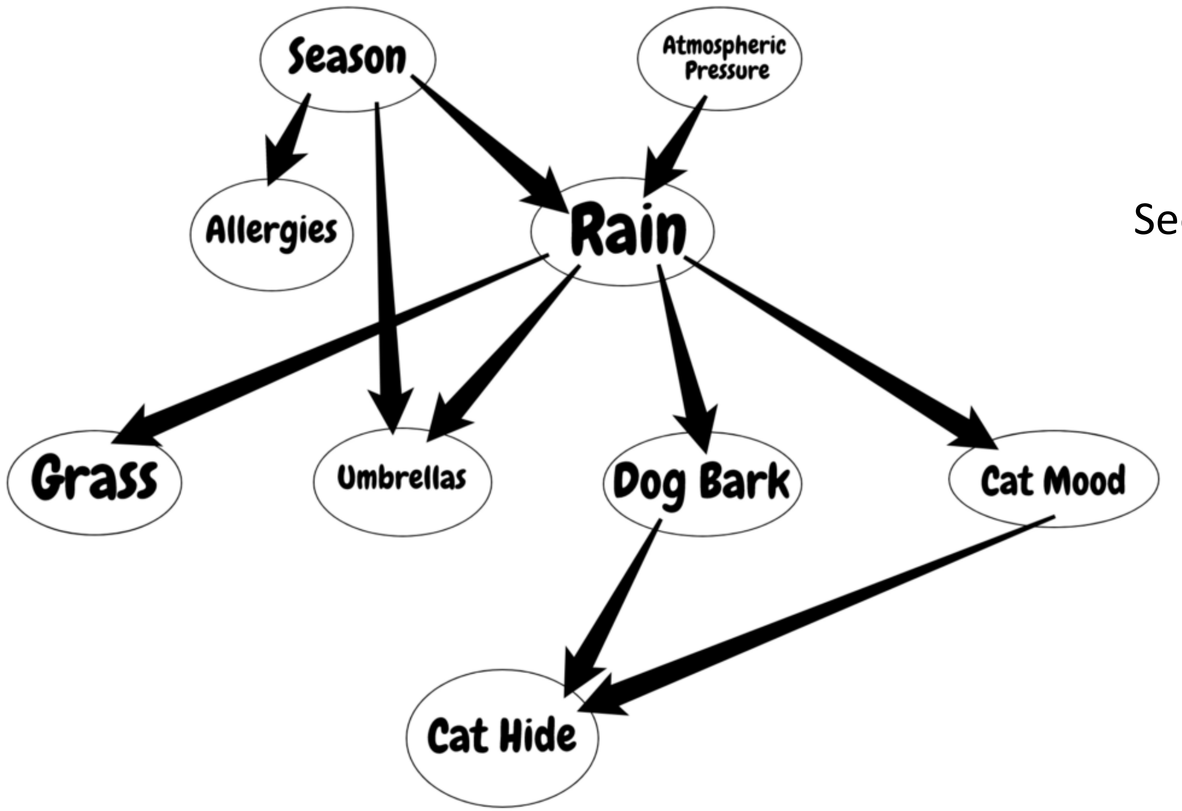


- Invert the probability unit (probit) model:

$$P(B|C = c) = 1 - P\left(C \leq \frac{c-\mu_C}{\sigma_C}\right)$$

Similar model is the inverse logistic (logit) function: $P(B|C = c) = 1 - \frac{1}{1+e^{s\cdot\frac{c-\mu_C}{\sigma_C}}}$ where *s* is the probit's mean

Another example



See 13.2.4 for a more complex case study

Example by Atakan Güney, towardsdatascience.com

Evaluating probability

- Let *X* represent what we want to know
- Let *e* represent one or more evidence values (things we measured)
- Recall that $P(X|e) = \frac{P(X,e)}{P(e)} = \alpha P(X, e)$ where $\alpha = {}^{1}/_{P(e)}$
- Let *y* be variables that are *latent* (hidden or unobservable) are denoted
- Then:

$$P(X|e) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y)$$

Note: $\sum_y$ sums over all combinations of the y latent variables

Evaluating probability

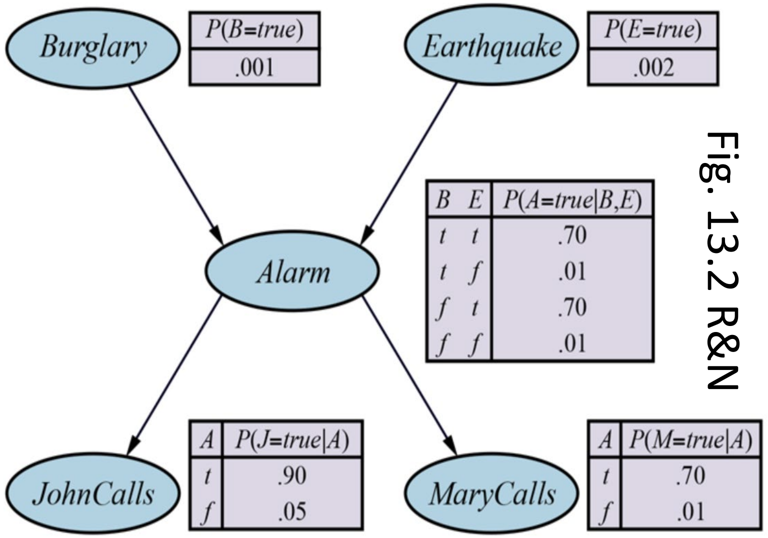


Fig. 13.2 R&N

- Let us consider the burglar alarm example
- Suppose we want to query:
  $P(Burglary|JohnCalls = True, MaryCalls = True))$

$$P(b|j,m) = \alpha \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$

naïve complexity for Booleans $O(n2^n)$

- the first two terms do not depend on a. Hence:

$$P(b|j,m) = \alpha \sum_e P(b)P(e) \sum_a P(a|b,e)P(j|a)P(m|a)$$

complexity for Booleans $O(2^n)$

Evaluating probability

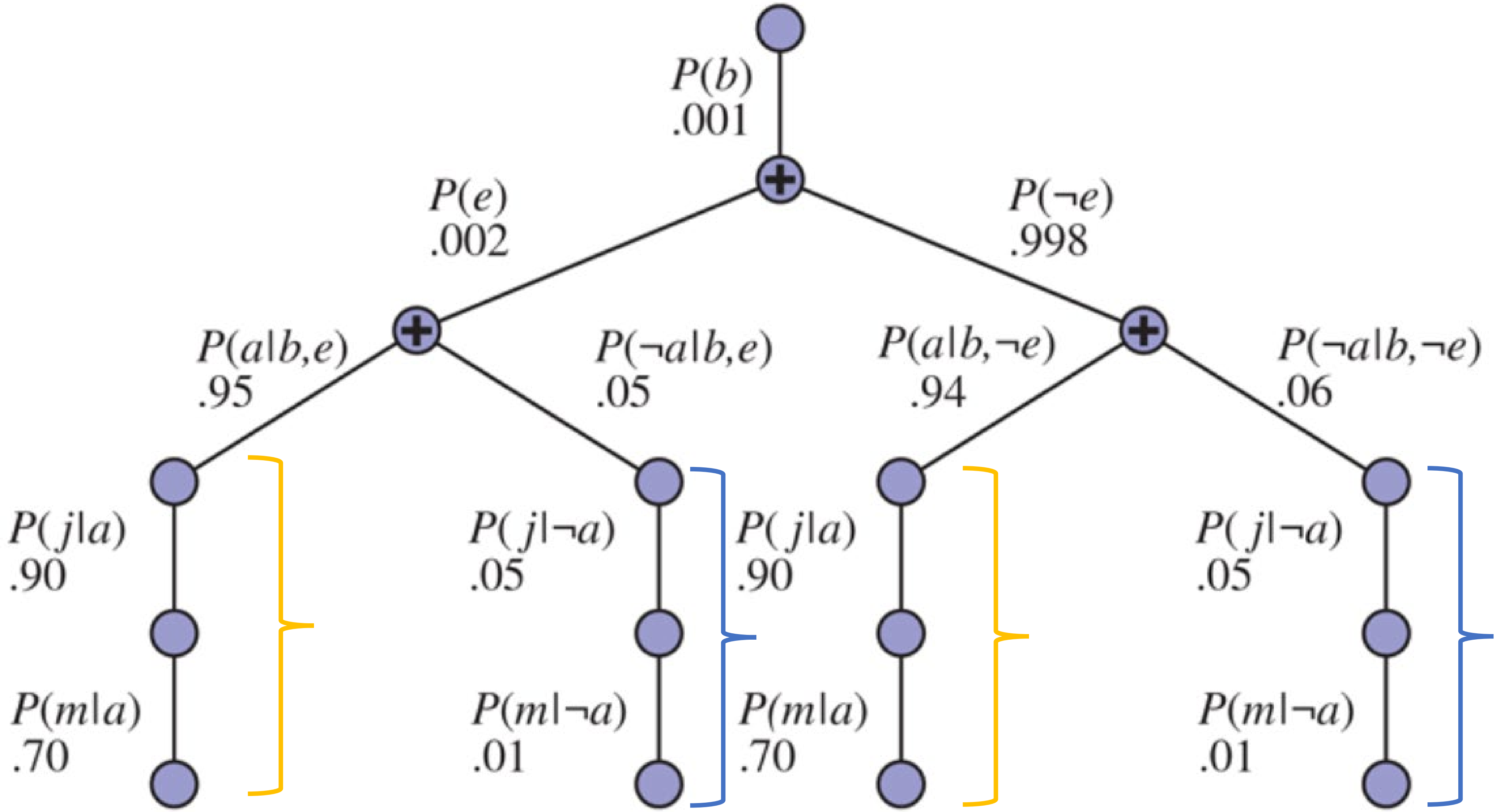
function ENUMERATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
      $\mathbf{e}$, observed values for variables $\mathbf{E}$
      bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is $\mathbf{e}$ extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in $\mathbf{e}$
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is $\mathbf{e}$ extended with $V = v$

Fig. 13.11 R&N

Redundancies



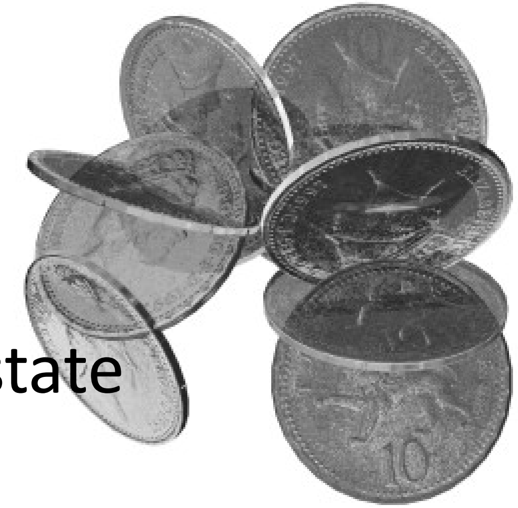
More efficient computation

- Evaluating common subgraphs once is more efficient and makes a difference in large graphs. Variable elimination algorithm (13.3.2) does this

- There are also approximate evaluation algorithms that are covered later in chapter 13

You are not responsible for these.

hidden Markov models



almostsuremath.com

- Used for modeling processes that have an unobservable state
- Example
  - 2 coins behind a screen with different odds of heads/tails
    Here we'll let one coin be fair, the other is biased
  - I have a process
    - Flip coin
    - Choose the next coin to flip
  - All we observe are sequences: H, H, H, H, T, H, T, T, H, …
- hidden Markov models let us model these types of systems

Markov property

- Let q_i be the state that we are in at time *i*. If $q_4 = fair$, we are using the fair coin for the 4th flip in our previous example.
- Chain rule states

$$\mathrm{P}(q_1, q_2, \ldots, q_T) = \mathrm{P}(q_1) \prod_{i=2}^{T} \mathrm{P}(q_i | q_{i-1} q_{i-2} \cdots q_1)$$

- Markov property specifies conditional independence after 1 step (can be generalized to N steps)

$$\mathrm{P}(q_i | q_{i-1} q_{i-2} \cdots q_1) = P(q_i | q_{i-1})$$

Observed Markov models

- Finite state machine with state transition probabilities:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,0} & a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix}$$

$$\sum_{j=0}^{4} a_{ij} = 1$$

$$\text{where } 0 \leq i \leq 4$$

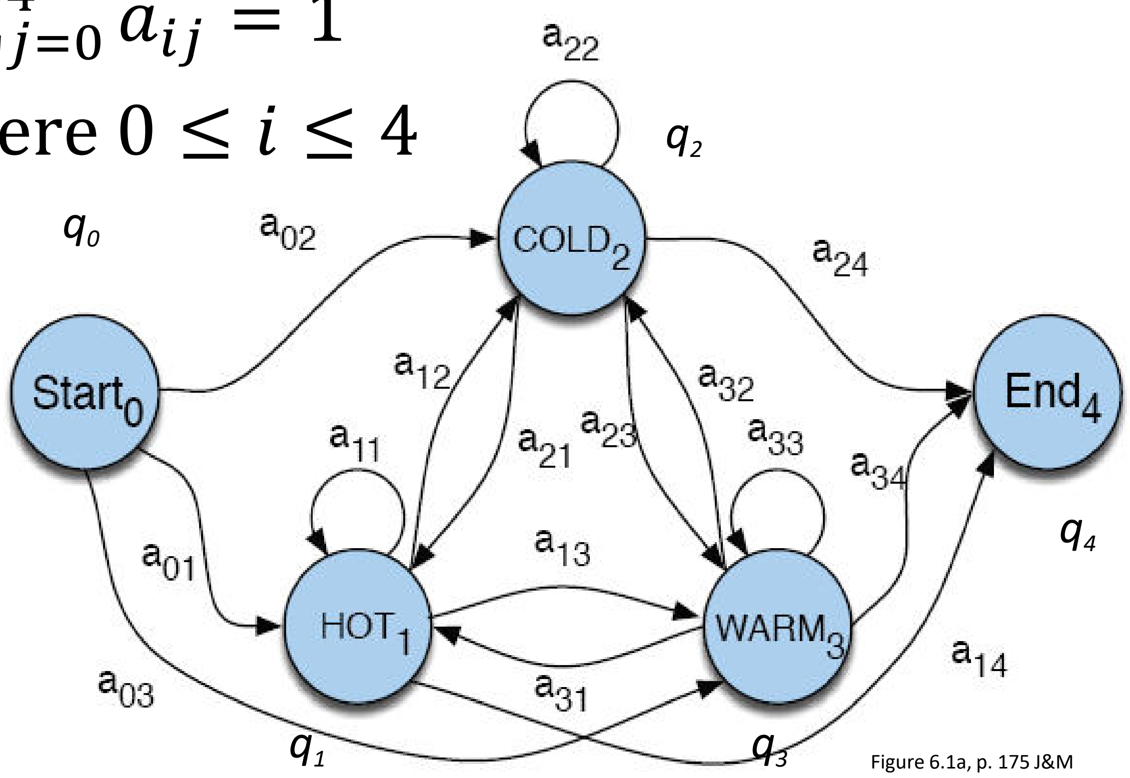


Figure 6.1a, p. 175 J&M

$a_{2,3}$ is the probability of moving from state 2 to state 3.

So the probability of going from state 2 at time 5 to state 3 at time 6 would be:

$$P(q_6 = 3 | q_5 = 2) = a_{2,3}$$

Observed Markov models

- Markov for a state sequence:

$$\mathrm{P}(q_i|q_1, \dots, q_{i-1}) = \mathrm{P}(q_i|q_{i-1})$$
e.g., $P(warm_3|hot_1) = a_{13}$

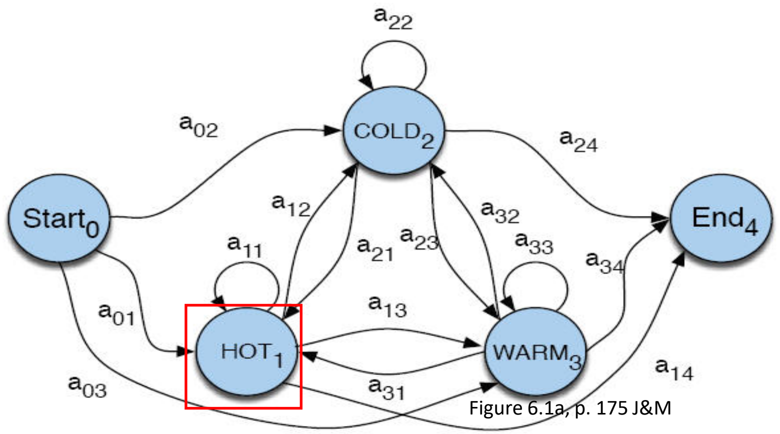


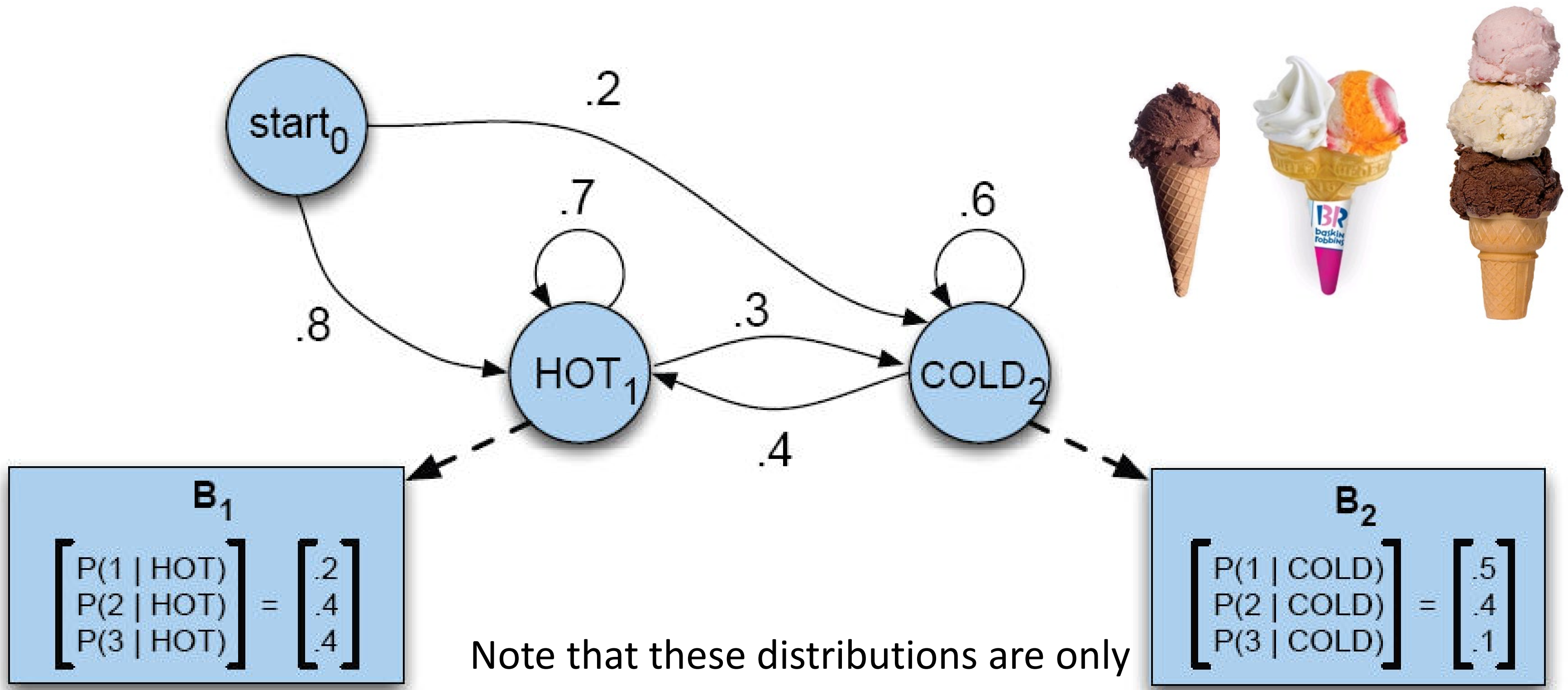
Figure 6.1a, p. 175 J&M

- Chain rule & Markov property

$$\mathrm{P}(q_1, q_2, \dots, q_T) = \mathrm{P}(q_1) \prod_{i=2}^{T} \mathrm{P}(q_i|q_{i-1})$$

State dependent distributions

Number of scoops of ice cream eaten daily



Note that these distributions are only dependent upon the state.

State-dependent transitions

- R&N present observation probabilities differently.
  - Each observation probability is written as an $S \times S\ matrix$

- We will use the notation of Rabiner's 1989 tutorial article in *Proc IEEE*

Notation worth remembering

- Observations (features)

$$O = \{o_1, o_2, \ldots, o_T\}$$

- Observations come from a discrete or continuous space and are independent of one another

$$o_i \in \Re^D \text{ or } o_i \in \{v_1, v_2, \ldots, v_{N_v}\}$$

- States $S = \{s_1, s_2, \ldots, s_{N_s}\}$
- State sequences[1]

$$q_1, q_2, \ldots, q_T \text{ where } q_i \in S$$

Markov chains

- Sequence can be seen as moving from one state to another, dependent only upon the previous state:
  - P(high | yesterday high, day before changing) = P(high | yesterday high)

Low pressure

Pressure changing

High pressure

State transition distribution

- Matrix A describes the state transition probabilities:

$$A = \begin{array}{c} \text{transition to} \\ \begin{array}{ccc} low & changing & high \end{array} \\ \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix} \end{array} \begin{array}{c} \text{transition from} \\ low \\ changing \\ high \end{array}$$

$$a_{ij} = \mathrm{P}(q_t = s_j | q_{t-1} = s_i)$$

Sometimes abbreviated as: $a_{ij} = \mathrm{P}(q_t = j | q_{t-1} = i)$

- P(high|low) = a_{13}=1/4

$$\sum_{j=1}^{N} a_{ij} = 1$$

Initial state distribution

- The Markov chain has a probability of starting in an initial state, denoted by the vector π

$$\pi = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{array}{c} low \\ changing \\ high \end{array}$$

- In this example, the starting state has a uniform (equally probable) distribution.

State-dependent distributions

- For each state s, there is a probability of seeing an observation o: $b_s(o)$
- For our weather model example:

$$b_1(o) = \begin{cases} \frac{3}{4} & o = \text{rain} \\ \frac{1}{4} & o = \text{sun} \end{cases} \qquad b_2(o) = \begin{cases} \frac{1}{2} & o = \text{rain} \\ \frac{1}{2} & o = \text{sun} \end{cases} \qquad b_3(o) = \begin{cases} \frac{1}{4} & o = \text{rain} \\ \frac{3}{4} & o = \text{sun} \end{cases}$$

What are the odds of that?

$$\mathrm{P}(O_1 = rain, Q_1 = changing, O_2 = sun, Q_2 = high, O_3 = sun,\ \mathrm{Q}_3 = high)$$

which we abbreviate as: $\mathrm{P}(o_1, q_1, o_2, q_2, o_3, q_3)$

by chaining Bayes rule $\mathrm{P}(AB) = \mathrm{P}(A \mid B)\Pr(B)\ldots$

$$= \mathrm{P}(\mathrm{q}_1)\mathrm{P}(\mathrm{o}_1 \mid q_1)\mathrm{P}(q_2 \mid q_1 o_1)\mathrm{P}(o_2 \mid q_2 q_1 o_1)\mathrm{P}(q_3 \mid o_2 q_2 q_1 o_1)\mathrm{P}(o_3 \mid q_3 o_2 q_2 o_1 q_1)$$

As o_i only dependent on state q_i

$$= \mathrm{P}(\mathrm{q}_1)\mathrm{P}(\mathrm{o}_1 \mid q_1)\mathrm{P}(q_2 \mid q_1 o_1)\mathrm{P}(o_2 \mid q_2)\mathrm{P}(q_3 \mid o_2 q_2 q_1 o_1)\mathrm{P}(o_3 \mid q_3)$$

and as we have a first order Markov chain

$$= \mathrm{P}(\mathrm{q}_1)\mathrm{P}(\mathrm{o}_1 \mid q_1)\mathrm{P}(q_2 \mid q_1)\mathrm{P}(o_2 \mid q_2)\mathrm{P}(q_3 \mid q_2)\mathrm{P}(o_3 \mid q_3)$$

$$= \pi_{changing} b_{changing}(rain) a_{changing,high} b_{high}(sun) a_{high,high} b_{high}(sun)$$

or using our state numbers: $\pi_2 b_2(rain) a_{2,3} b_3(sun) a_{3,3} b_3(sun)$

s_1
Low pressure

s_2
Pressure changing

s_3
High pressure

The *hidden* in hidden Markov model

- Barometer let us observe the state.

- Suppose we cannot observe state.

- Many state sequences are possible,
  each sequence has a probability of occurrence.

HMM

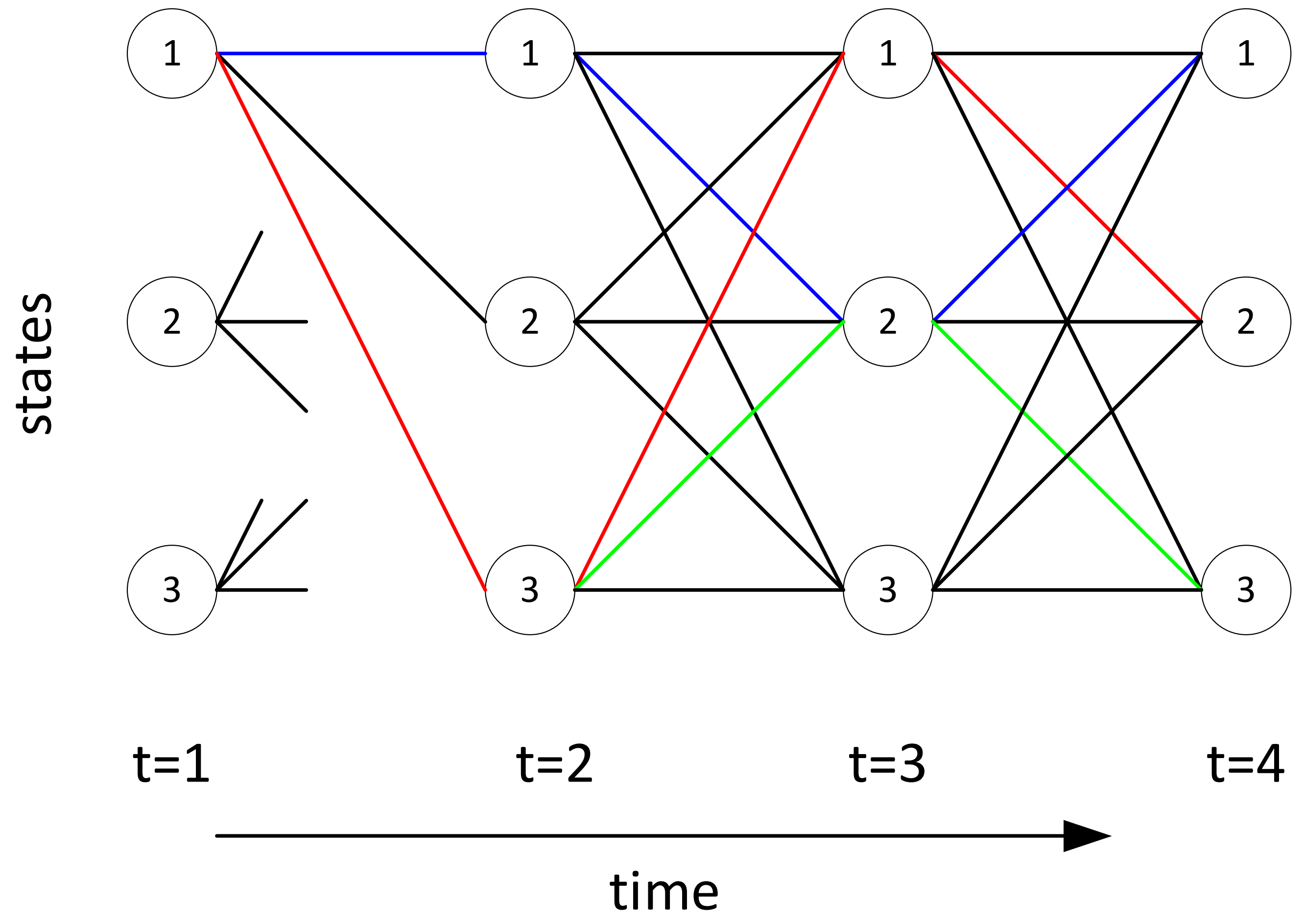
- Let $\Phi(A,B,\pi)$ denote a HMM where:
  - A – NxN state transition matrix. a_{ij} denotes the probability of transitioning from state i to j.
  - B – $\{b_j(k)\}$ – Set of state-dependent probability distributions. $1 \leq j \leq N_s$, k in O
  - π – Initial state distribution. π_j is the probability of starting in state j. $1 \leq j \leq N_s$

Top 3 List for HMMs

1. What is the probability of a given sequence O given model Φ?
2. What state sequence is most likely to account for O in model Φ?
3. How can we improve the parameters of Φ to better account for O?

Probability Evaluation

- Must evaluate all paths through model



Naive approach is exponential!

Probability evaluation

- Dynamic programming can be used
- Two algorithms
  - Forward procedure

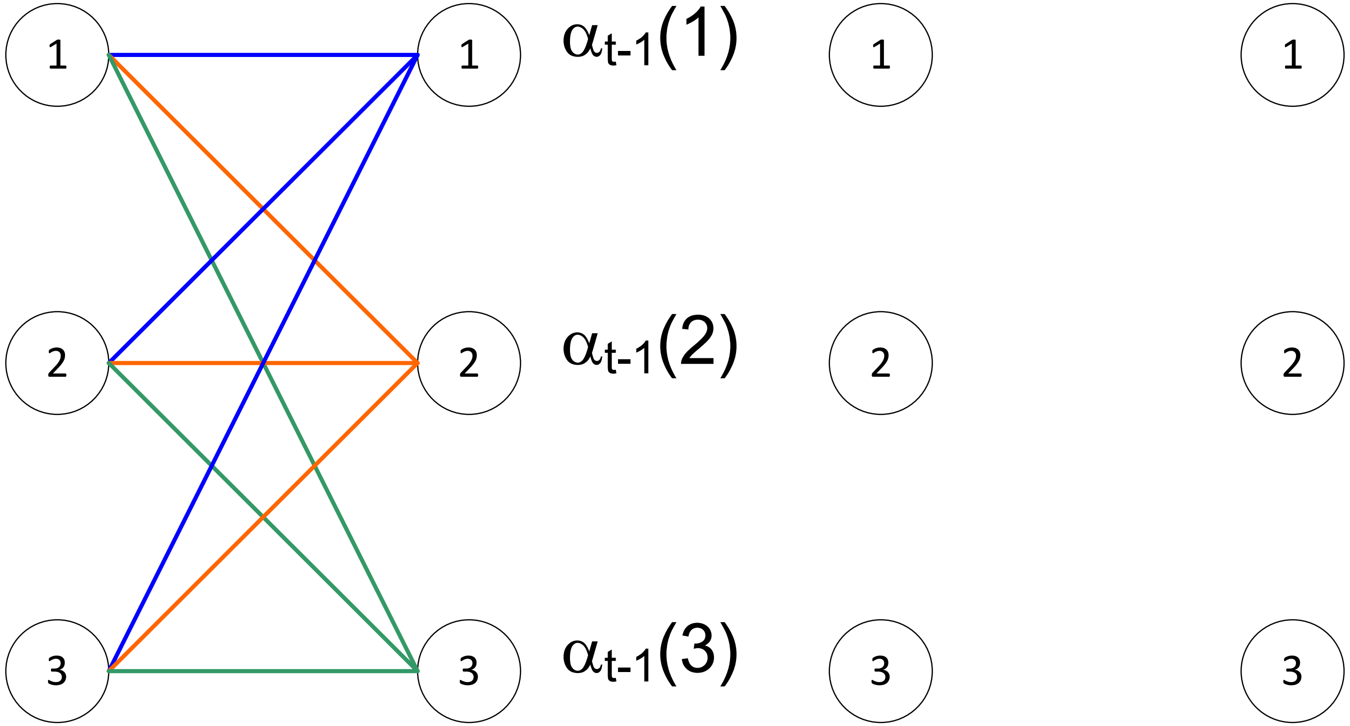
$$\alpha_t(i) = \mathrm{P}(\, o_1, o_2, \dots, o_t, q_t = s_i | \Phi)$$

  - Backward procedure

$$\beta_t(i) = \mathrm{P}(\, o_{t+1}, o_{t+2}, \dots, o_T, q_t = s_i | \Phi)$$

Forward algorithm

- Suppose we know the sum of all paths leading into each state j at time t-1:

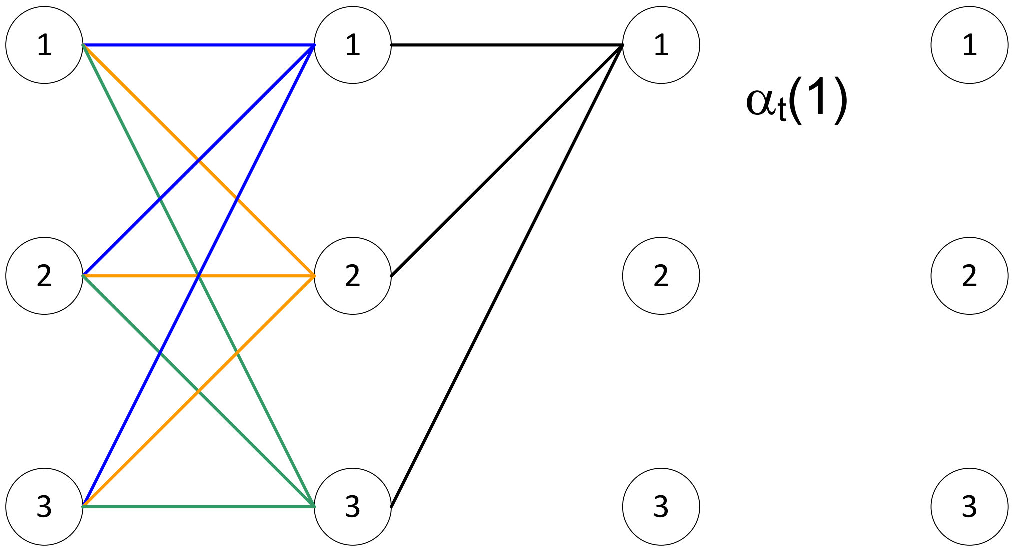


$$\alpha_{t-1}(j) = \mathrm{P}(\, o_1, o_2, \dots, o_{t-1}, q_{t-1} = s_j | \Phi)$$

Forward algorithm

- Then we can compute the probability of all paths leading into time t.

$$\alpha_t(i) = \mathrm{P}(o_1, o_2, \ldots, o_t, q_t = s_i \mid \Phi)$$



Forward trellis

Forward algorithm - $O(\mathrm{N}^2\mathrm{T})$

- Initialization

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad 1 \le i \le N$$

- Induction

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(o_t) \qquad \begin{array}{l} 1 \le j \le N \\ 2 \le t \le T \end{array}$$

- Termination

$$\mathrm{P}(O|\Phi) = \sum_{i=1}^{N} \alpha_T(i)$$

Optimal State Sequence

- The forward algorithm finds all paths through a model.
- Sometimes, we are interested in the best path through the model:
  - Perhaps we are interested in determining which states are associated with which observations.
  - Frequently, most of the paths contribute very little to the overall probability.

Viterbi algorithm: Determine optimal state sequence

- Another example of dynamic programming
- Finds the most likely path through the model
- Similar to the forward algorithm
  - Uses max instead of sum
  - Keeps extra information about the best path

Viterbi Algorithm

- Initialization
  - Probability to start in state i

$$V_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N$$

  - $B_t(i)$ – The previous state which transitioned into state i at time t-1. (0 indicates no previous state.)

$$B_1(i) = 0 \qquad 1 \leq i \leq N$$

Viterbi Algorithm *O(N²T)*

- Induction
  - Find the best path leading into the current state and account for the probability of the observation

  - Record the previous node on the best path

$$V_t(j) = \max_{1 \le i \le N} \left[V_{t-1}(i) a_{ij} \right] b_j(o_t) \qquad \begin{array}{l} 1 \le j \le N \\ 2 \le t \le T \end{array}$$

$$B_t(j) = s_{\underset{1 \le i \le N}{\operatorname{argmax}} [V_{t-1}(i) a_{ij}]} \qquad \begin{array}{l} 1 \le j \le N \\ 2 \le t \le T \end{array}$$

Viterbi Algorithm

- Termination

$$\mathrm{P}^{BestPath}(X) = \max_{1 \le i \le N}[V_T(i)]$$

$$q_T^{BestPath} = s_{\arg\max_{1 \le i \le N}[V_T(i)]}$$

- Extracting the best path:

$$\text{for t} = \mathrm{T}-1, \mathrm{T}-2, \ \ldots, 1$$
$$q_t^{BestPath} = s_{B_{t+1}(q_{t+1}^{BestPath})}$$

Log Domain Viterbi Implementation

- Operates in the log probability domain
- Multiplications are replaced with addition
- Not covered in text.

Parameter Estimation

- Application of the Expectation-Maximization (EM) algorithm
  - If we had all the information
    - true state sequence
    - observations
  - then techniques such as maximum-likelihood estimation could be used to improve our parameter set

EM algorithm

- Problem: Some information is unknown

- Solution:
  - Use the expectation operator to determine the expected values of missing parameters
  - Determine new parameters
  - Repeat

EM Algorithm

- Guaranteed to converge to a local maximum
- For speech applications, no more than 5-15 iterations are typically required
- For HMMs, the resulting formulae are known as the Baum-Welch reestimation equations.

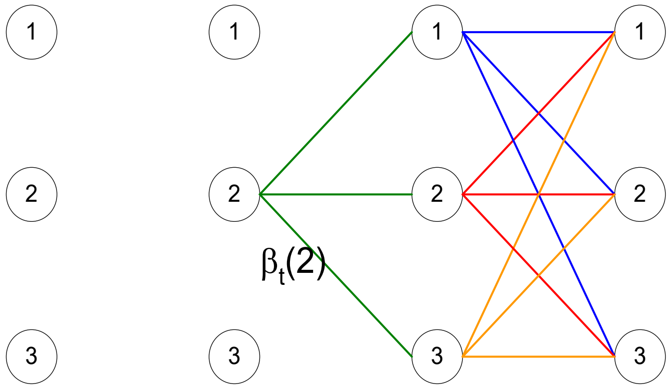
Some needed concepts

- The backward algorithm – similar to the forward algorithm, but works from T down towards 1.
- $\gamma_t(i,j)$ – Given a model and observation sequence, the probability of transitioning from state i at t-1 to j at t given the model Φ and acoustic evidence O

$$P(q_{t-1} = s_i, q_t = s_j | O, \Phi)$$

Backward algorithm

$$\beta_t(i) = \mathrm{P}(o_{t+1}, o_{t+2}, \ldots, o_T \mid q_t = s_i, \Phi)$$



Note: β_t(i) does not include the probability of observing o_t.

Backward Algorithm $O(N^2T)$

- Initialization

$$\beta_T(i) = 1 \qquad 1 \le i \le N$$

- Induction

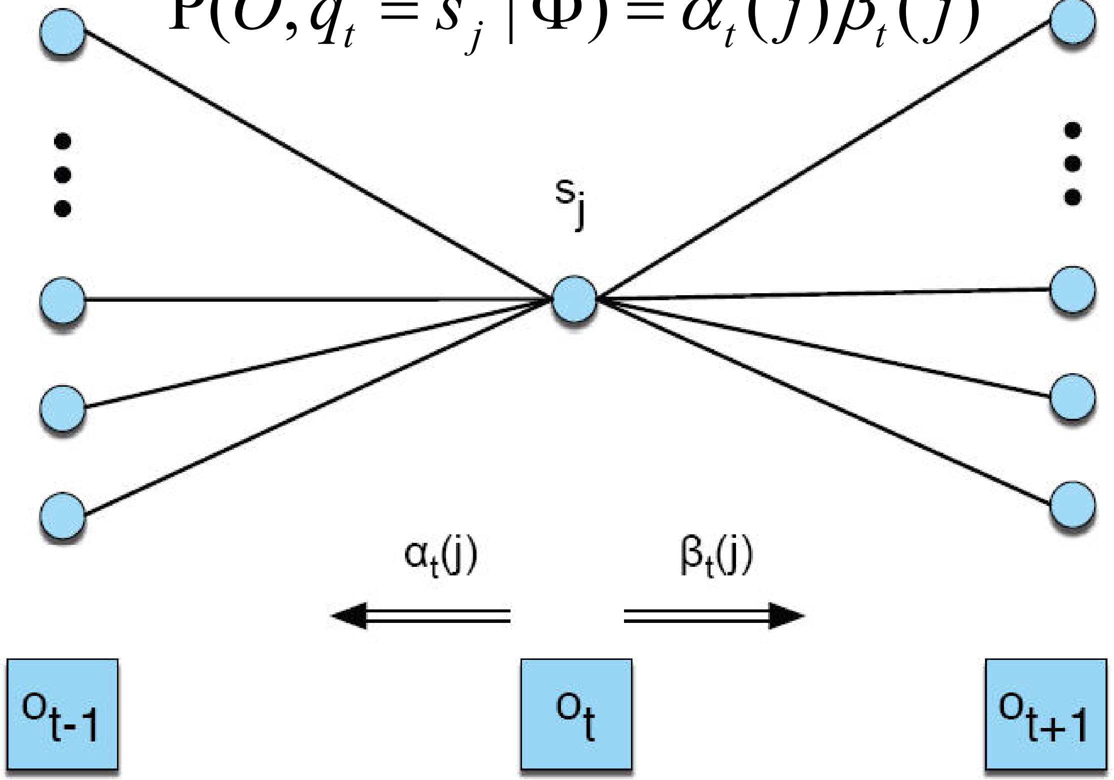
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j\,(o_{t+1})\beta_{t+1}(j) \qquad \begin{array}{l} 1 \le j \le N \\ 1 \le t \le T-1 \end{array}$$

- Termination not needed, but possible

Forward-Backward relationship

- $\alpha_t(i)$ = all paths into q_t=s_i and observing o_1, o_2, ..., o_t.
- $\beta_t(i)$ = all paths out of q_t=s_j and observing o_{t+1},o_{t+2}, ..., o_T.

$$P(O, q_t = s_j \mid \Phi) = \alpha_t(j)\beta_t(j)$$

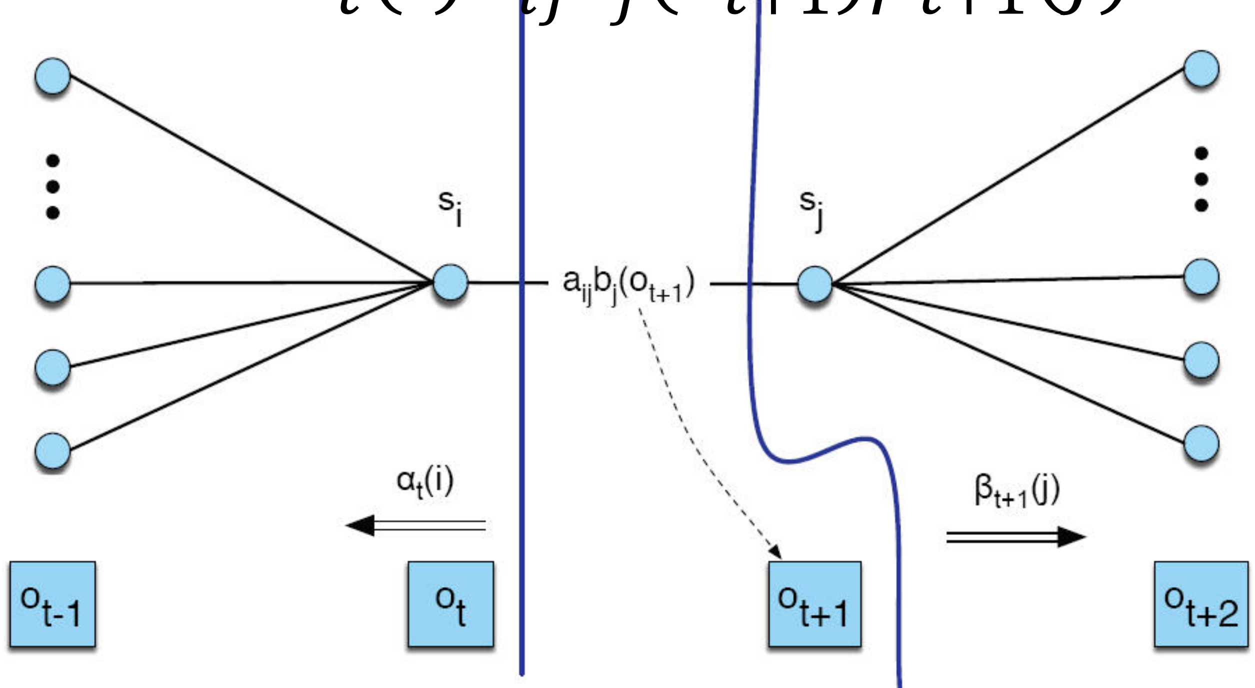


J&M p. 191

Constrained path probability

Suppose we want the probability of all paths through a specific transition from t-1 to t:

$$P(O, q_t = s_i, q_{t+1} = s_j | \Phi) \\ = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$



J&M p. 190

$\gamma_t(i,j)$ Probability state i to j at time t

$$\gamma_t(i,j) \triangleq \mathrm{P}(\,q_{t-1} = s_i, q_t = s_j | O, \Phi)$$

$$= \frac{\mathrm{P}(\,q_{t-1} = s_i, s_t = s_j, O|\Phi)}{P(O|\Phi)}$$

Bayes rule

$$= \frac{\alpha_{t-1}(i) a_{ij} b_j(o_t) \beta_t(j)}{\sum_{k=1}^{N} \alpha_T(k)}$$

previous slide, and

$$P(O|\Phi) = \sum_{k=1}^{N} \alpha_T(k)$$

Special case: $\gamma_1(i,j)$

- Calls for non-existant transition between q_0 and q_1. We define $\alpha_0(i)$=1 and a_{ij} at time 0 as $\pi_{j:}$

$$\gamma_1(i,j) \quad = \frac{\alpha_0(i) a_{ij} b_j(o_1)\beta_1(j)}{\sum_{k=1}^{N} \alpha_T(k)}$$

$$= \frac{\pi_j b_j(o_1)\beta_1(j)}{\sum_{k=1}^{N} \alpha_T(k)}$$

Baum-Welch equations

- The EM algorithm can be used to derive the Baum-Welch reestimation equations.
- Computation of the expectations is done with the γ function.
- Once the expectation has been computed, a maximum likelihood estimate can be computed for the model.

Initial state distribution

- Percentage of time that we will be in state i at time t=1

$$\hat{\pi}_i = \sum_{k=1}^{N} \gamma_1(i,k)$$

- For many speech applications, we desire a starting state. In this case π_{start}=1. The reestimation formulas will not change this.

State transition

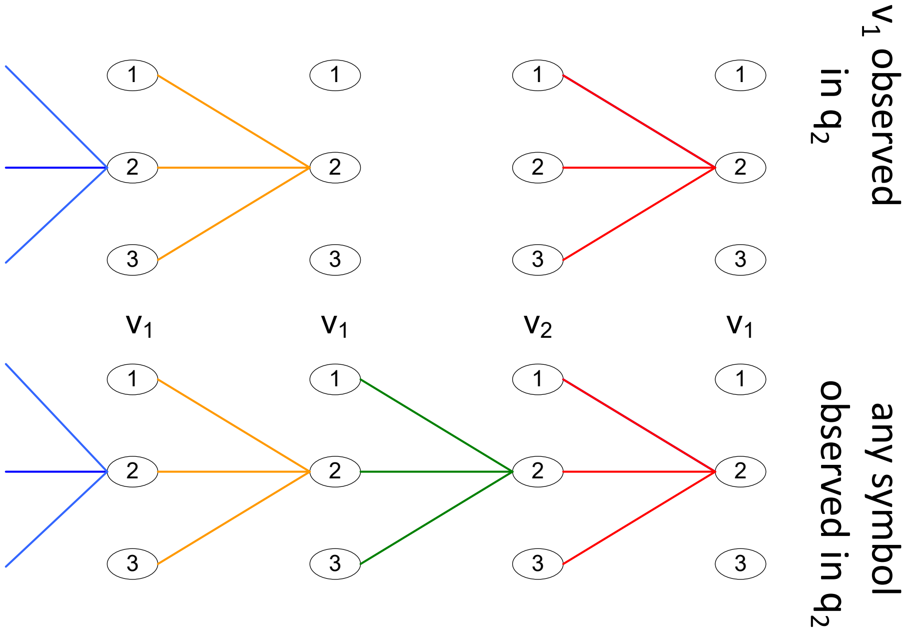
- We can think of this as the expected number of transitions from state i to j divided by the expected number of all transitions:

$$\hat{a}_{ij} = \frac{\sum_{t=2}^{T} \gamma_t(i,j)}{\sum_{t=2}^{T} \sum_{k=1}^{N} \gamma_t(i,k)}$$

Note: Huang, Acero, and Hon sum from 1 to N (eq. 8.40, p 392) which includes the initial state probability π_i. Most authors do not do this.

State-dependent pdfs

- For each time where symbol v_k is seen, we need to determine the probability of seeing v_k given that we are in state s_j.
- We divide this expectation by the expected probability of any symbol given sate s_j.



State-dependent pdfs

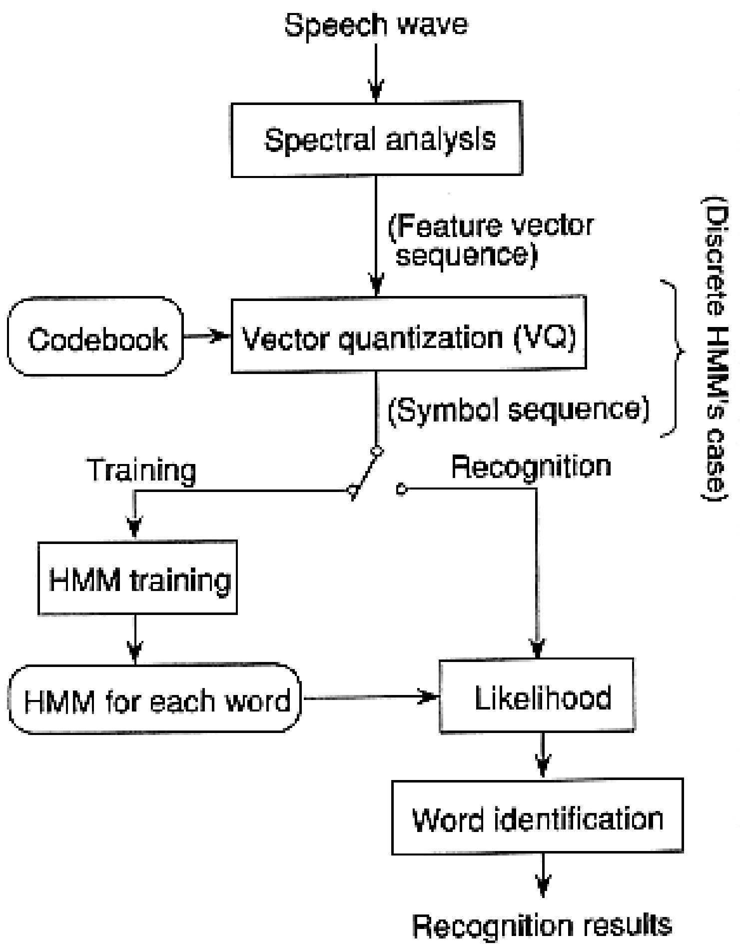
- Divide the expected number of transitions into state j where symbol o_k occurs by all transitions into state j:

$$\hat{b}_j(k) = \frac{\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma_t(i,j)\delta(o_t,v_k)}{\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma_t(i,j)}$$

where $\delta(o,v) \triangleq \begin{cases} 1 & o = v \\ 0 & o \neq v \end{cases}$

$$= \frac{\sum_{t \text{ such that } o_t = v_k} \sum_{i=1}^{N}\gamma_t(i,j)}{\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma_t(i,j)}$$

Isolated Word Recognizer



Furui 2001, p. 284

Vector quantization is the same think as k-means. We map a continuous vector to a discrete symbol.