Introduction to Graphical Models[1]

David Barber

University College London

[1] All notes and software available from http://www.cs.ucl.ac.uk/staff/d.barber/brml/

Table of Contents

Section 1

Probability

Probability

Why Probability?

- Probability is a logical calculus of uncertainty.
- Natural framework to use in models of physical systems, such as the Ising Model (1920) and in AI applications, such as the HMM (Baum 1966, Stratonovich 1960).

The need for structure

- We often want to make a probabilistic description of many objects (electron spins, neurons, customers, *etc.*).
- Typically the representational and computational cost of probabilistic models grows exponentially with the number of objects represented.
- Without introducing strong structural limitations about how these objects can interact, probability is a non-starter.
- For this reason, computationally 'simpler' alternatives (such as fuzzy logic) were introduced to try to avoid some of these difficulties – however, these are typically frowed on by purists.

Graphical Models

- We can use graphs to represent how objects can probabilistically interact with each other.
- Graphical Models and then a marriage between Graph and Probability theory.
- Many of the quantities that we would like to compute in a probability distribution can then be related to operations on the graph.
- The computational complexity of operations can often be related to the structure of the graph.
- Graphical Models are now used as a standard framework in Engineering, Statistics and Computer Science.

Rules of probability

$p(x = \mathsf{x})$: the probability of variable x being in state x.

$$p(x = \mathsf{x}) = \begin{cases} 1 & \text{we are certain } x \text{ is in state x} \\ 0 & \text{we are certain } x \text{ is not in state x} \end{cases}$$

Values between 0 and 1 represent the degree of certainty of state occupancy.

---

domain

$\mathrm{dom}(x)$ denotes the states x can take. For example, $\mathrm{dom}(c) = \{\mathsf{heads}, \mathsf{tails}\}$. When summing over a variable $\sum_x f(x)$, the interpretation is that all states of x are included, *i.e.* $\sum_x f(x) \equiv \sum_{\mathsf{s} \in \mathrm{dom}(x)} f(x = \mathsf{s})$.

---

distribution

Given a variable, x, its domain $\mathrm{dom}(x)$ and a full specification of the probability values for each of the variable states, $p(x)$, we have a distribution for x.

---

normalisation

The summation of the probability over all the states is 1:

$$\sum_{x \in \mathrm{dom}(x)} p(x = \mathsf{x}) = 1$$

We will usually more conveniently write $\sum_x p(x) = 1$.

Operations

AND

Use the shorthand $p(x, y) \equiv p(x \cap y)$ for $p(x$ and $y)$. Note that $p(y, x) = p(x, y)$.

---

marginalisation

Given a joint distr. $p(x, y)$ the marginal distr. of x is defined by

$$p(x) = \sum_y p(x, y)$$

More generally,

$$p(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = \sum_{x_i} p(x_1, \ldots, x_n)$$

Conditional Probability and Bayes' Rule

The probability of event x conditioned on knowing event y (or more shortly, the probability of x given y) is defined as

$$p(x|y) \equiv \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)} \quad \text{(Bayes' rule)}$$

Throwing darts

$$p(\text{region } 5|\text{not region } 20) = \frac{p(\text{region } 5, \text{not region } 20)}{p(\text{not region } 20)}$$
$$= \frac{p(\text{region } 5)}{p(\text{not region } 20)} = \frac{1/20}{19/20} = \frac{1}{19}$$

Interpretation

$p(A = \text{a}|B = \text{b})$ should not be interpreted as 'Given the event $B = \text{b}$ has occurred, $p(A = \text{a}|B = \text{b})$ is the probability of the event $A = \text{a}$ occurring'. The correct interpretation should be '$p(A = \text{a}|B = \text{b})$ is the probability of A being in state a under the constraint that B is in state b'.

Probability tables

The a priori probability that a randomly selected Great British person would live in England, Scotland or Wales, is 0.88, 0.08 and 0.04 respectively.

We can write this as a vector (or probability table) :

$$\begin{pmatrix} p(Cnt = \mathsf{E}) \\ p(Cnt = \mathsf{S}) \\ p(Cnt = \mathsf{W}) \end{pmatrix} = \begin{pmatrix} 0.88 \\ 0.08 \\ 0.04 \end{pmatrix}$$

whose component values sum to 1.

The ordering of the components in this vector is arbitrary, as long as it is consistently applied.

Probability tables

We assume that only three Mother Tongue languages exist : English (Eng), Scottish (Scot) and Welsh (Wel), with conditional probabilities given the country of residence, England (E), Scotland (S) and Wales (W). Using the state ordering:

$$MT = [\text{Eng}, \text{Scot}, \text{Wel}]; \qquad Cnt = [\text{E}, \text{S}, \text{W}]$$

we write a (fictitious) conditional probability table

$$p(MT|Cnt) = \begin{pmatrix} 0.95 & 0.7 & 0.6 \\ 0.04 & 0.3 & 0.0 \\ 0.01 & 0.0 & 0.4 \end{pmatrix}$$

Probability tables

The distribution $p(Cnt, MT) = p(MT|Cnt)p(Cnt)$ can be written as a 3×3 matrix with (say) rows indexed by country and columns indexed by Mother Tongue:

$$\begin{pmatrix} 0.95 \times 0.88 & 0.7 \times 0.08 & 0.6 \times 0.04 \\ 0.04 \times 0.88 & 0.3 \times 0.08 & 0.0 \times 0.04 \\ 0.01 \times 0.88 & 0.0 \times 0.08 & 0.4 \times 0.04 \end{pmatrix} = \begin{pmatrix} 0.836 & 0.056 & 0.024 \\ 0.0352 & 0.024 & 0 \\ 0.0088 & 0 & 0.016 \end{pmatrix}$$

By summing a column, we have the marginal

$$p(Cnt) = \begin{pmatrix} 0.88 \\ 0.08 \\ 0.04 \end{pmatrix}$$

Summing the rows gives the marginal

$$p(MT) = \begin{pmatrix} 0.916 \\ 0.0592 \\ 0.0248 \end{pmatrix}$$

Probability tables

Large numbers of variables

For joint distributions over a larger number of variables, $x_i, i = 1, \ldots, D$, with each variable x_i taking K_i states, the table describing the joint distribution is an array with $\prod_{i=1}^{D} K_i$ entries.

Explicitly storing tables therefore requires space exponential in the number of variables, which rapidly becomes impractical for a large number of variables.

Indexing

A probability distribution assigns a value to each of the joint states of the variables. For this reason, $p(T, J, R, S)$ is considered equivalent to $p(J, S, R, T)$ (or any such reordering of the variables), since in each case the joint setting of the variables is simply a different index to the same probability.

One should be careful not to confuse the use of this indexing type notation with functions $f(x, y)$ which are in general dependent on the variable order.

Independence

Variables x and y are independent if knowing one event gives no extra information about the other event. Mathematically, this is expressed by

$$p(x, y) = p(x)p(y)$$

Independence of x and y is equivalent to

$$p(x|y) = p(x) \Leftrightarrow p(y|x) = p(y)$$

If $p(x|y) = p(x)$ for all states of x and y, then the variables x and y are said to be independent. We write then $x \perp\!\!\!\perp y$.

---

interpretation

Note that $x \perp\!\!\!\perp y$ doesn't mean that, given y, we have no information about x. It means the only information we have about x is contained in $p(x)$.

---

factorisation

If

$$p(x, y) = kf(x)g(y)$$

for some constant k, and positive functions $f(\cdot)$ and $g(\cdot)$ then x and y are independent.

Conditional Independence

$$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$$

denotes that the two sets of variables $\mathcal{X}$ and $\mathcal{Y}$ are independent of each other given the state of the set of variables $\mathcal{Z}$. This means that

$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) = p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z}) \text{ and } p(\mathcal{X}|\mathcal{Y}, \mathcal{Z}) = p(\mathcal{X}|\mathcal{Z})$$

for all states of $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. In case the conditioning set is empty we may also write $\mathcal{X} \perp\!\!\!\perp \mathcal{Y}$ for $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \emptyset$, in which case $\mathcal{X}$ is (unconditionally) independent of $\mathcal{Y}$.

Conditional independence does not imply marginal independence

$$p(x, y) = \sum_z \underbrace{p(x|z)p(y|z)}_{\text{cond. indep.}} p(z) \neq \underbrace{\sum_z p(x|z)p(z)}_{p(x)} \underbrace{\sum_z p(y|z)p(z)}_{p(y)}$$

Conditional dependence

If $\mathcal{X}$ and $\mathcal{Y}$ are not conditionally independent, they are conditionally dependent. This is written

$$\mathcal{X} \top\!\!\!\top \mathcal{Y} | \mathcal{Z}$$

Conditional Independence example

Based on a survey of households in which the husband and wife each own a car, it is found that:

$$\text{wife's car type} \perp\!\!\!\perp \text{husband's car type} \mid \text{family income}$$

There are 4 car types, the first two being 'cheap' and the last two being 'expensive'. Using w for the wife's car type and h for the husband's:

$$p(w|inc = \text{low}) = \begin{pmatrix} 0.7 \\ 0.3 \\ 0 \\ 0 \end{pmatrix}, \quad p(w|inc = \text{high}) = \begin{pmatrix} 0.2 \\ 0.1 \\ 0.4 \\ 0.3 \end{pmatrix}$$

$$p(h|inc = \text{low}) = \begin{pmatrix} 0.2 \\ 0.8 \\ 0 \\ 0 \end{pmatrix}, \quad p(h|inc = \text{high}) = \begin{pmatrix} 0 \\ 0 \\ 0.3 \\ 0.7 \end{pmatrix}$$

$$p(inc = \text{low}) = 0.9$$

Conditional Independence example

Then the marginal distribution $p(w, h)$ is

$$p(w, h) = \sum_{inc} p(w|inc)p(h|inc)p(inc)$$

giving

$$p(w, h) = \begin{pmatrix} 0.126 & 0.504 & 0.006 & 0.014 \\ 0.054 & 0.216 & 0.003 & 0.007 \\ 0 & 0 & 0.012 & 0.028 \\ 0 & 0 & 0.009 & 0.021 \end{pmatrix}$$

From this we can find the marginals and calculate

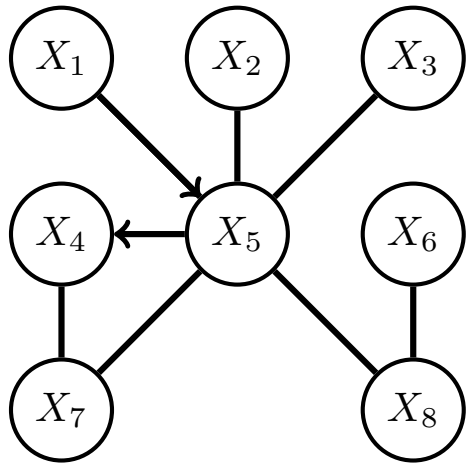
$$p(w)p(h) = \begin{pmatrix} 0.117 & 0.468 & 0.0195 & 0.0455 \\ 0.0504 & 0.2016 & 0.0084 & 0.0196 \\ 0.0072 & 0.0288 & 0.0012 & 0.0028 \\ 0.0054 & 0.0216 & 0.0009 & 0.0021 \end{pmatrix}$$

This shows that whilst $w \perp\!\!\!\perp h|\, inc$, it is not true that $w \perp\!\!\!\perp h$. For example, even if we don't know the family income, if we know that the husband has a cheap car then his wife must also have a cheap car – these variables are therefore dependent.

Section 2

Graphs

Graphs



Definition

A graph consists of nodes (vertices) and undirected or directed links (edges) between nodes.

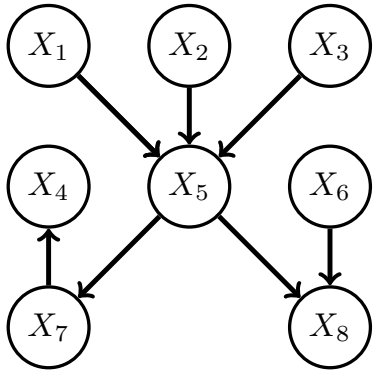
Path

A path from X_i to X_j is a sequence of connected nodes starting at X_i and ending at X_j.

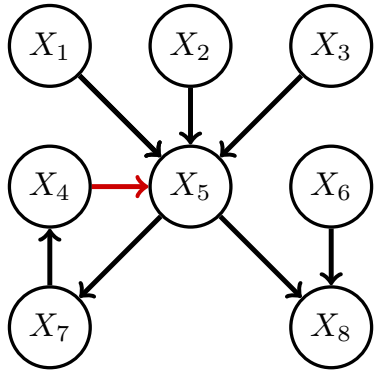
Directed Graphs

All the edges are directed:

Directed Acyclic Graph



Directed Cyclic Graph



DAG

Directed Acyclic Graph: Graph in which by following the direction of the arrows a node will never be visited more than once.

Parents and Children:

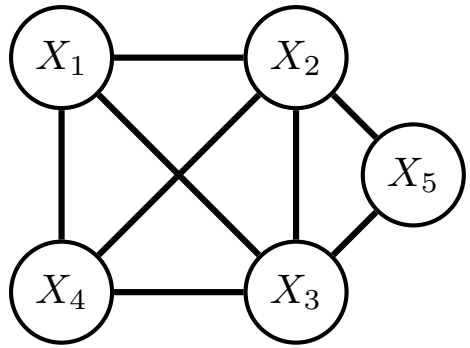
X_i is a parent of X_j if there is a link from X_i to X_j. X_i is a child of X_j if there is a link from X_j to X_i.

Ancestors and Descendants:

The ancestors of a node X_i are the nodes with a directed path ending at X_i. The descendants of X_i are the nodes with a directed path beginning at X_i.

Undirected Graph

All the edges are undirected:



Clique

A clique is a fully connected subset of nodes. (X_1, X_2, X_4) forms a (non-maximal) clique.

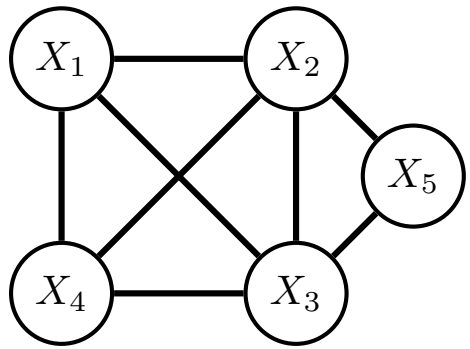
Maximal Clique

Clique which is not a subset of a larger clique. (X_1, X_2, X_3, X_4) and (X_2, X_3, X_5) are both maximal cliques.

Connectivity

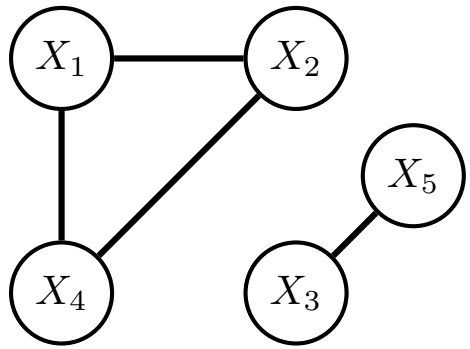
Connected graph

There is a path between every pair of vertices:



Connected components

In a non-connected graph, the connected components are the connected-subgraphs:

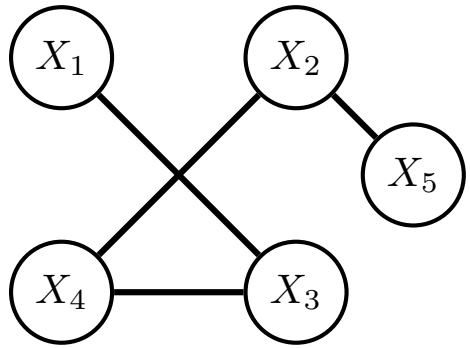


(X_1, X_2, X_4) and (X_3, X_5) are the two connected components.

Connectedness

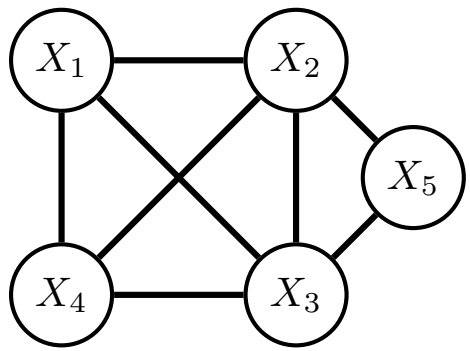
Singly-connected

There is only one path from any node a to another other node b



Multiply-connected

A graph is multiply-connected if it is not singly-connected:



Section 3

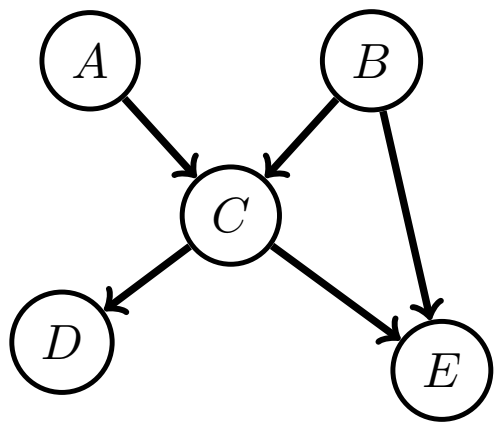
Directed Graphical Models

Belief Networks (Bayesian Networks)

A belief network is a directed acyclic graph in which each node has associated the conditional probability of the node given its parents.

The joint distribution is obtained by taking the product of the conditional probabilities:

$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|C)p(E|B, C)$$



$p(E|B, C)$

Example – Part I

Sally's burglar **A**larm is sounding. Has she been **B**urgled, or was the alarm triggered by an **E**arthquake? She turns the car **R**adio on for news of earthquakes.

Choosing an ordering

Without loss of generality, we can write

$$\begin{aligned} p(A,R,E,B) &= p(A|R,E,B)p(R,E,B) \\ &= p(A|R,E,B)p(R|E,B)p(E,B) \\ &= p(A|R,E,B)p(R|E,B)p(E|B)p(B) \end{aligned}$$

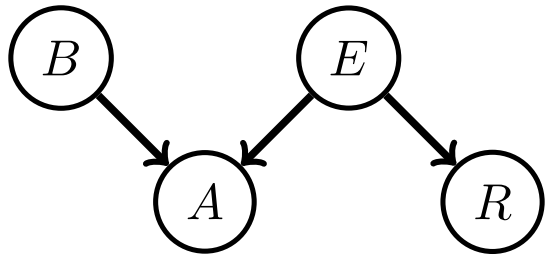
Assumptions:

- The alarm is not directly influenced by any report on the radio, $p(A|R,E,B) = p(A|E,B)$
- The radio broadcast is not directly influenced by the burglar variable, $p(R|E,B) = p(R|E)$
- Burglaries don't directly 'cause' earthquakes, $p(E|B) = p(E)$

Therefore

$$p(A,R,E,B) = p(A|E,B)p(R|E)p(E)p(B)$$

Example – Part II: Specifying the Tables



$p(A|B,E)$

Alarm $= 1$	Burglar	Earthquake
0.9999	1	1
0.99	1	0
0.99	0	1
0.0001	0	0

$p(R|E)$

Radio $= 1$	Earthquake
1	1
0	0

The remaining tables are $p(B = 1) = 0.01$ and $p(E = 1) = 0.000001$. The tables and graphical structure fully specify the distribution.

Example Part III: Inference

Initial Evidence: The alarm is sounding

$$p(B=1|A=1) = \frac{\sum_{E,R} p(B=1,E,A=1,R)}{\sum_{B,E,R} p(B,E,A=1,R)}$$
$$= \frac{\sum_{E,R} p(A=1|B=1,E)p(B=1)p(E)p(R|E)}{\sum_{B,E,R} p(A=1|B,E)p(B)p(E)p(R|E)} \approx 0.99$$

Additional Evidence: The radio broadcasts an earthquake warning:

A similar calculation gives $p(B=1|A=1,R=1) \approx 0.01$.

Initially, because the alarm sounds, Sally thinks that she's been burgled. However, this probability drops dramatically when she hears that there has been an earthquake.

The earthquake 'explains away' to an extent the fact that the alarm is ringing.

Examples of Belief Networks in Machine Learning

Prediction (discriminative)

$p(class|input)$

Prediction (generative)

$p(class|input) \propto p(input|class)p(class)$

Time-series

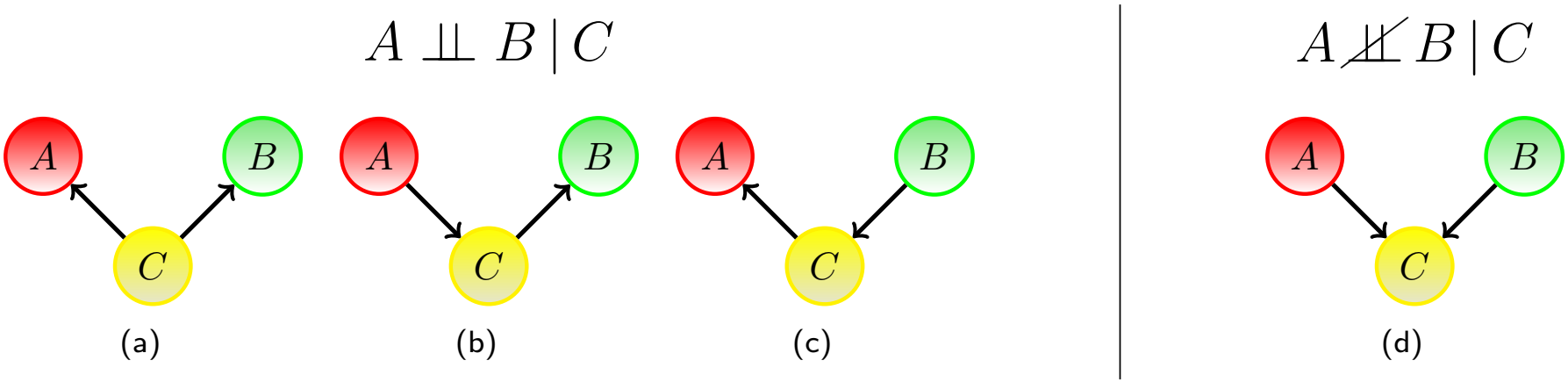
Markov chains, Hidden Markov Models.

Unsupervised learning

$p(data) = \sum_{latent} p(data|latent)p(latent)$.

Independence ⫫ in Belief Networks – Part I

All belief networks with three nodes and two links:



- In (a), (b) and (c), A, B are conditionally independent given C.

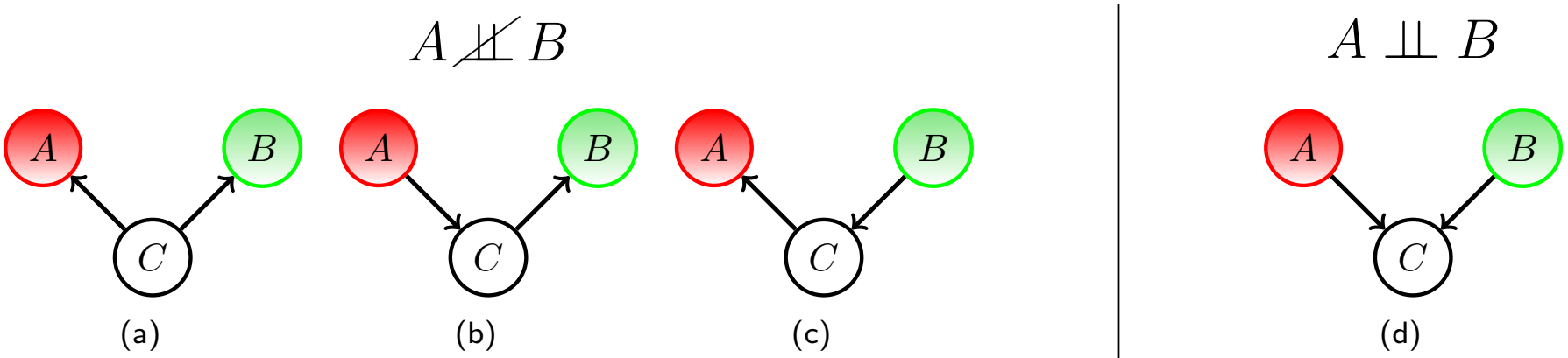
  (a) $p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(A|C)p(B|C)p(C)}{p(C)} = p(A|C)p(B|C)$

  (b) $p(A,B|C) = \frac{p(A)p(C|A)p(B|C)}{p(C)} = \frac{p(A,C)p(B|C)}{p(C)} = p(A|C)p(B|C)$

  (c) $p(A,B|C) = \frac{p(A|C)p(C|B)p(B)}{p(C)} = \frac{p(A|C)p(B,C)}{p(C)} = p(A|C)p(B|C)$

- In (d) the variables A, B are conditionally dependent given C, $p(A,B|C) \propto p(C|A,B)p(A)p(B)$.

Independence $\perp\!\!\!\perp$ in Belief Networks – Part II

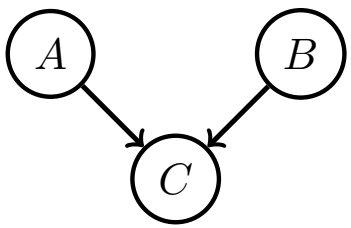


- In (a), (b) and (c), the variables A, B are marginally dependent.
- In (d) the variables A, B are marginally independent.

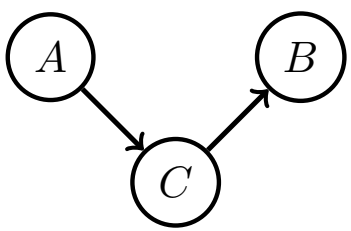
$$p(A,B) = \sum_C p(A,B,C) = \sum_C p(A)p(B)p(C|A,B) = p(A)p(B)$$

Collider

A collider contains two or more incoming arrows along a chosen path.
Summary of two previous slides:



If C has more than one incoming link, then $A \perp\!\!\!\perp B$ and $A \not\perp\!\!\!\perp B \,|\, C$. In this case C is called collider.

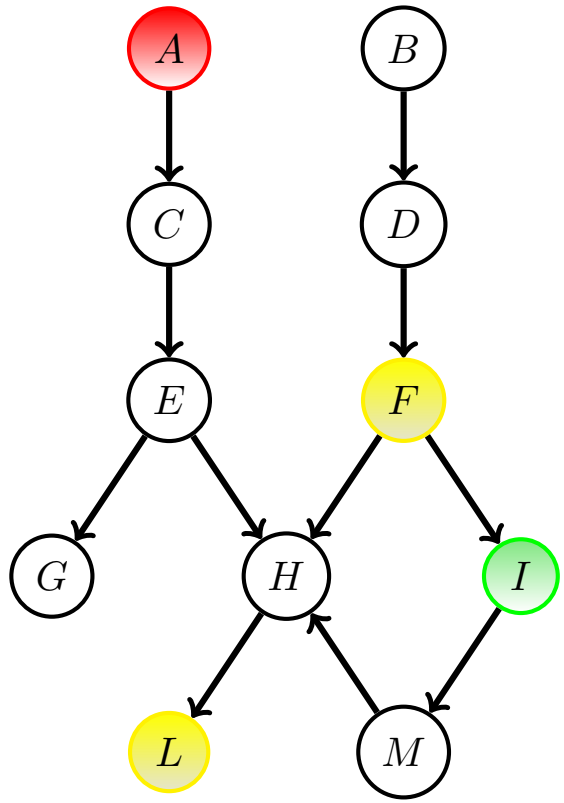


If C has at most one incoming link, then $A \perp\!\!\!\perp B \,|\, C$ and $A \not\perp\!\!\!\perp B$. In this case C is called non-collider.

Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

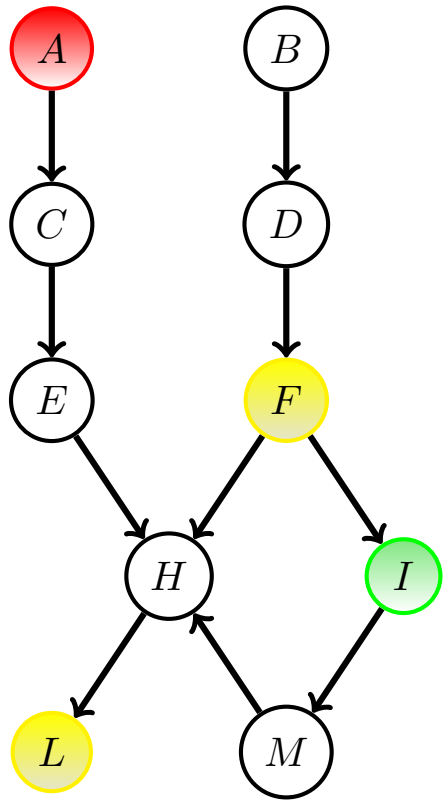
- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \,|\, \mathcal{Z}$.



Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

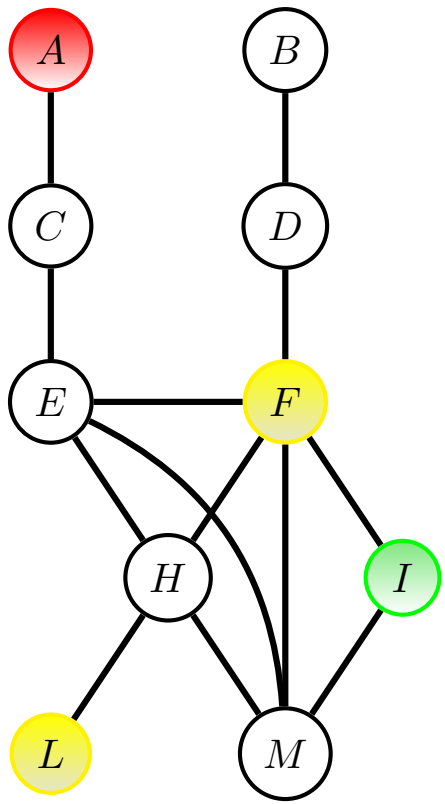
- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$.



Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

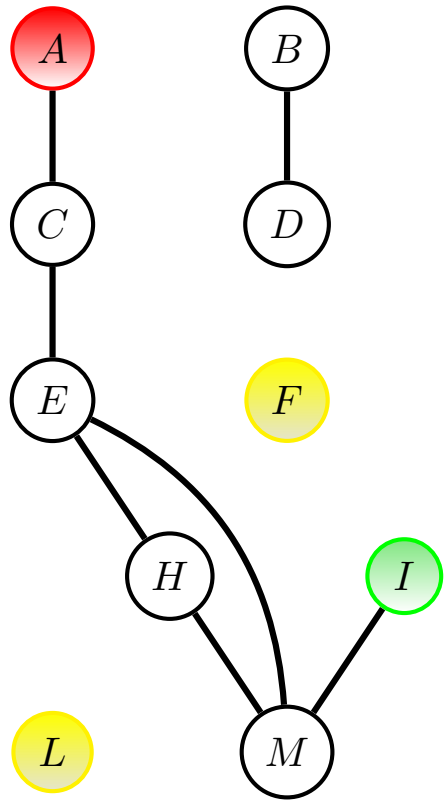
- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \,|\, \mathcal{Z}$.



Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \,|\, \mathcal{Z}$.



$A \top\!\!\!\top I | F, L$

Section 4

Inference in Trees

Inference

Inference corresponds to using the distribution to answers questions about the environment.

---

examples

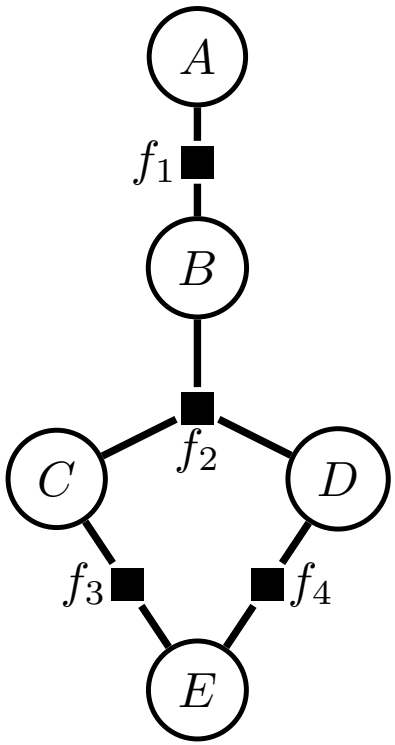
- What is the probability $p(x = 4|y = 1, z = 2)$?
- What is the most likely joint state of the distribution $p(x, y)$?
- What is the entropy of the distribution $p(x, y, z)$?
- What is the probability that this example is in class 1?
- What is the probability the stock market will do down tomorrow?

Computational Efficiency

- Inference can be computationally very expensive and we wish to characterise situations in which inferences can be computed efficiently.
- For singly-connected graphical models, and certain inference questions, there (usually) exist efficient algorithms based on the concept of message passing.
- In general, the case of multiply-connected models is computationally inefficient.

Factor Graphs

A square node represents a factor (non negative function) of its neighbouring variables.



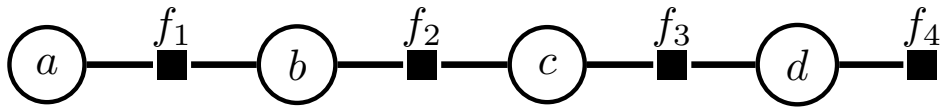
The joint function is the product of all factors:

$$f(A, B, C, D, E) = f_1(A, B) f_2(B, C, D) f_3(C, E) f_4(D, E)$$

Factor graphs are useful for performing efficient computations (not just for probability).

Sum-Product Algorithm - Non Branching Tree

$p(a,b,c,d) \propto f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$ $\quad a,b,c,d$ binary variables

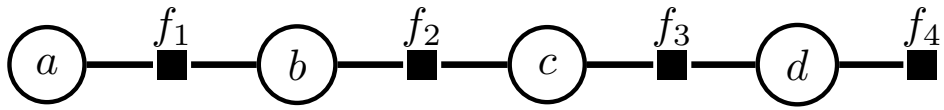


$$p(a) = \sum_{b,c,d} p(a,b,c,d)$$

$$\propto \sum_{b,c,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d) \Rightarrow 2^3 \text{ sums}$$

Sum-Product Algorithm - Non Branching Tree

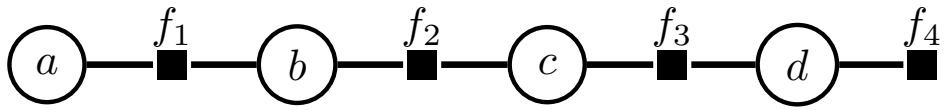
$p(a,b,c,d) \propto f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$ $\quad a,b,c,d$ binary variables



$$
\begin{aligned}
p(a) &= \sum_{b,c,d} p(a,b,c,d) \\
&\propto \sum_{b,c,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d) \Rightarrow 2^3 \text{ sums} \\
&= \sum_{b} f_1(a,b) \sum_{c} f_2(b,c) \sum_{d} f_3(c,d)\, f_4(d) \Rightarrow 2 \times 3 \text{ sums}
\end{aligned}
$$

Sum-Product Algorithm - Non Branching Tree

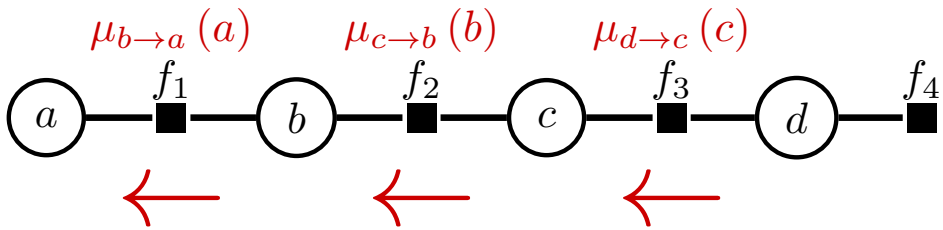
$p(a,b,c,d) \propto f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$ $\quad a,b,c,d$ binary variables



$$
\begin{aligned}
p(a) &= \sum_{b,c,d} p(a,b,c,d) \\
&\propto \sum_{b,c,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d) \\
&= \sum_{b} f_1(a,b) \underbrace{\sum_{c} f_2(b,c) \underbrace{\sum_{d} f_3(c,d)\, f_4(d)}_{\mu_{d\to c}(c)}}_{\mu_{c\to b}(b)} \\
&\qquad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\mu_{b\to a}(a)}
\end{aligned}
$$

Sum-Product Algorithm - Non Branching Tree

$p(a,b,c,d) \propto f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$ $\quad a,b,c,d$ binary variables



Passing variable-to-variable messages from d up to a

$$p(a) = \sum_{b,c,d} p(a,b,c,d)$$

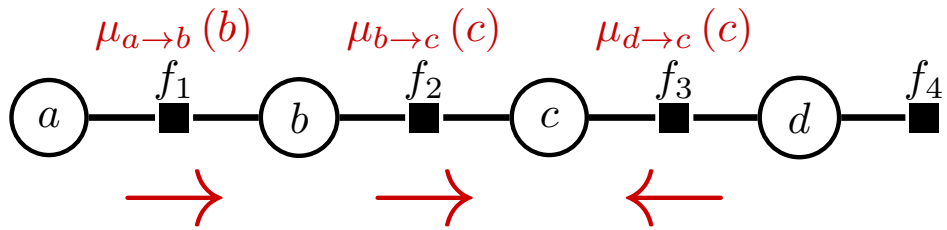
$$\propto \sum_{b,c,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$$

$$= \sum_b f_1(a,b) \underbrace{\sum_c f_2(b,c) \underbrace{\sum_d f_3(c,d)\, f_4(d)}_{\mu_{d\to c}(c)}}_{\mu_{c\to b}(b)}$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{\mu_{b\to a}(a)}$$

Sum-Product Algorithm - Non Branching Tree

For $p(c)$ need to send messages in both directions

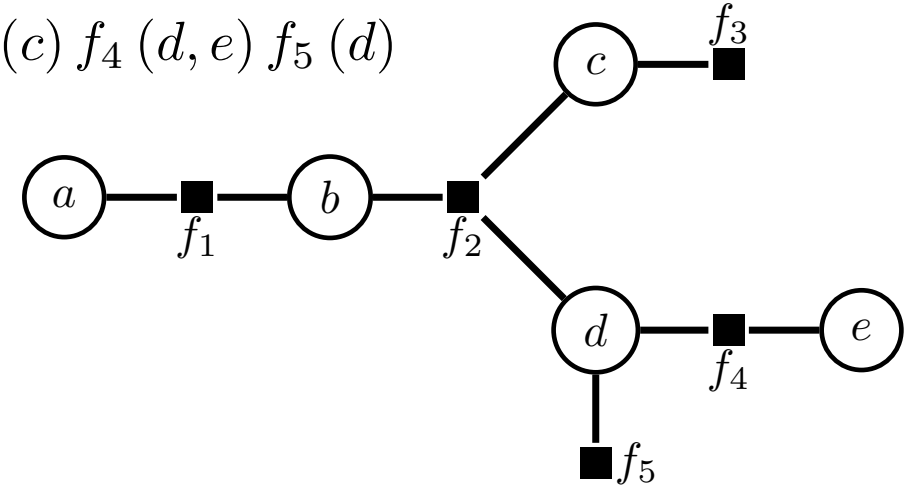


$$p(c) \propto \sum_{a,b,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$$

$$= \underbrace{\sum_b \underbrace{\sum_a f_1(a,b)}_{\mu_{a\to b}(b)}\, f_2(b,c)}_{\mu_{b\to c}(c)} \underbrace{\sum_d f_3(c,d)\, f_4(d)}_{\mu_{d\to c}(c)}$$

Sum-Product Algorithm – Branching Tree

$$p(a,b,c,d,e) \propto f_1(a,b)\, f_2(b,c,d)\, f_3(c)\, f_4(d,e)\, f_5(d)$$



Define factor-to-variable messages and variable-to-factor messages

$$p(a) \propto \sum_b f_1(a,b) \sum_{c,d} f_2(b,c,d) \underbrace{f_3(c)}_{\mu_{c\to f_2}(c)=\mu_{f_3\to c}(c)} \underbrace{f_5(d)}_{\mu_{f_5\to d}(d)} \underbrace{\sum_e f_4(d,e)}_{\mu_{f_4\to d}(d)}$$

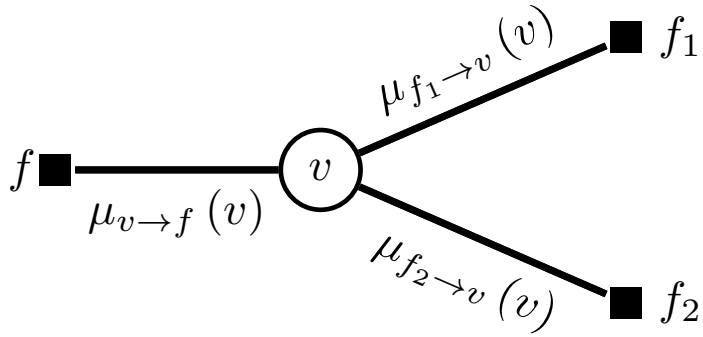
$$\underbrace{\underbrace{\underbrace{\mu_{f_5\to d}(d)\,\mu_{f_4\to d}(d)}_{\mu_{d\to f_2}(d)}}_{\mu_{b\to f_1}(b)=\mu_{f_2\to b}(b)}}_{\mu_{f_1\to a}(a)}$$

$\Rightarrow$ **Marginal inference for a singly-connected structure is 'easy'.**

Sum-Product Algorithm for Factor Graphs

Variable to factor message

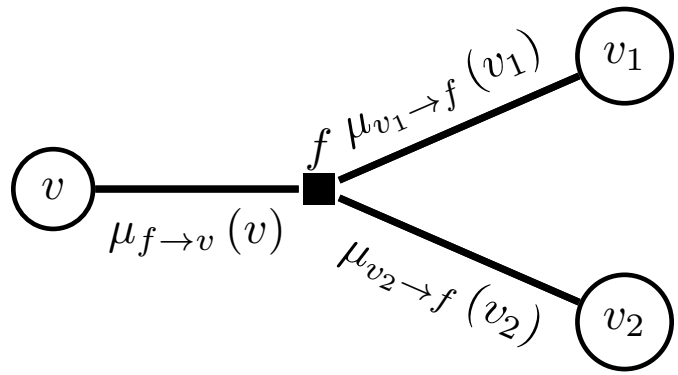
$$\mu_{v \to f}(v) = \prod_{f_i \sim v \setminus f} \mu_{f_i \to v}(v)$$



Messages from extremal variables are set to 1

Factor to variable message

$$\mu_{f \to v}(v) = \sum_{\{v_i\}} f(v, \{v_i\}) \prod_{v_i \sim f \setminus v} \mu_{v_i \to f}(v_i)$$



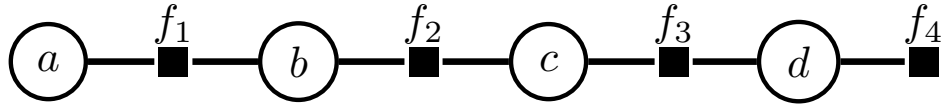
Messages from extremal factors are set to the factor

Marginal

$p(v) \propto \prod_{f_i \sim v} \mu_{f_i \to v}(v)$

Max Product algorithm

$p(a,b,c,d) \propto f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$ $\quad a,b,c,d$ binary variables



$$\max_{a,b,c,d} p(a,b,c,d) = \max_{a,b,c,d} f_1(a,b)\, f_2(b,c)\, f_3(c,d)\, f_4(d)$$

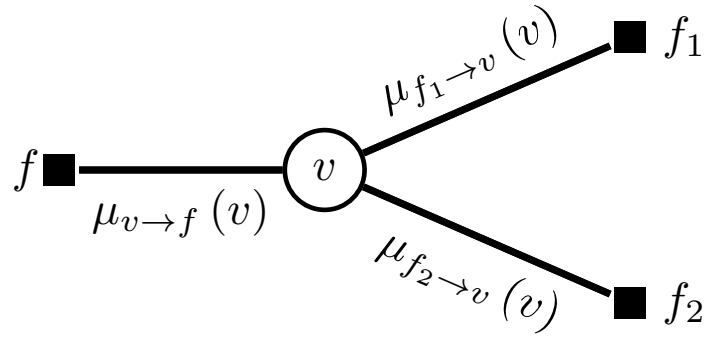
$$= \max_a \max_b f_1(a,b) \underbrace{\max_c f_2(b,c) \underbrace{\max_d f_3(c,d)\, f_4(d)}_{\mu_{d\to c}(c)}}_{\mu_{c\to b}(b)}$$

$$\mu_{b\to a}(a)$$

Max Product Algorithm for Factor Graphs

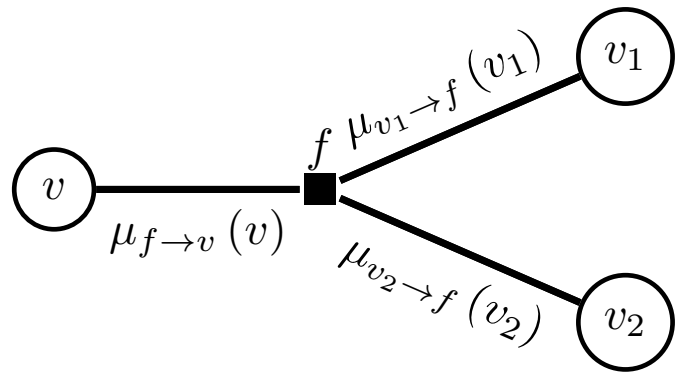
Variable to factor message

$$\mu_{v \to f}(v) = \prod_{f_i \sim v \setminus f} \mu_{f_i \to v}(v)$$



Factor to variable message

$$\mu_{f \to v}(v) = \max_{\{v_i\}} f(v, \{v_i\}) \prod_{v_i \sim f \setminus v} \mu_{v_i \to f}(v_i)$$



Most probable state (of joint)

$$v^* = \operatorname*{argmax}_v \prod_{f_i \sim v} \mu_{f_i \to v}(v)$$

Message Passing

- Also known as ‘belief propagation’ or ‘dynamic programming’.
- Note that for non-branching graphs (they look like ‘lines’), only variable to variable messages are required.
- For message passing to work we need to be able to distribute the operator over the factors (which means that the operator algebra is a semiring) and that the graph is singly-connected.
- Provided the above conditions hold, ‘marginal’ inference scales linearly with the number of nodes in the graph.

Message Passing

- If the graph is multiply-connected, message passing can still be implemented since it is a local algorithm. This is a popular approximation technique.
- Sometimes it is possible to identify a singly-connected structure from a multiply-connected structure by conditioning on a small set of variables (the cut-set). One can then run a set of message-passing algorithms, one for each state of the cut-set.
- What if the operator algebra is not a semiring? Won't work in general. An example is where we want

$$\max_{c,e,f} \sum_{a,b,d} p(a,b,c,d,e,f)$$

In this case, the $\max \sum$ operator is not distributive (the max of a sum is not the same as the sum of a max).

Computational Tractability

- Loosely speaking, singly-connected graphical models are easy to work with – the computational effort typically scales linearly with the number of variables in the distribution.
- Multiply-connected graphs (apart from special cases) are generally computationally hard to deal with.
- Many models of natural systems correspond to multiply-connected graphs (due to space being having dimension greater than 1).

Section 5

Markov Models

Time-Series

A time-series is an ordered sequence:

$$x_{a:b} = \{x_a, x_{a+1}, \ldots, x_b\}$$

So that one can consider the 'past' and 'future' in the sequence. The x can be either discrete or continuous.

Biology

Gene sequences. Emphasis is on understanding sequences, filling in missing values, clustering sequences, detecting patterns. Hidden Markov Models are one of the key tools in this area.

Finance

Price movement prediction.

Planning

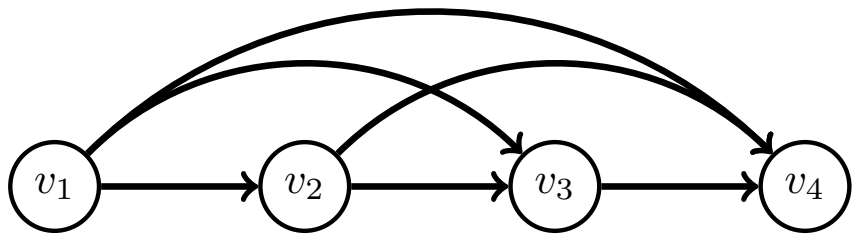
Forecasting – eg how many newspaper to deliver to retailers.

Markov Models

For timeseries data $v_1, \ldots, v_T$, we need a model $p(v_{1:T})$. For causal consistency, it is meaningful to consider the decomposition

$$p(v_{1:T}) = \prod_{t=1}^{T} p(v_t | v_{1:t-1})$$

with the convention $p(v_t|v_{1:t-1}) = p(v_1)$ for $t = 1$.



Independence assumptions

It is often natural to assume that the influence of the immediate past is more relevant than the remote past and in Markov models only a limited number of previous observations are required to predict the future.

Markov Chain

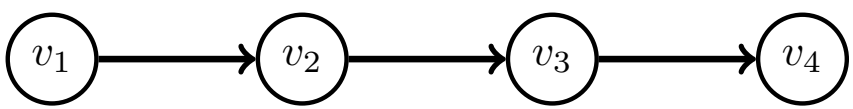
Only the recent past is relevant:

$$p(v_t|v_1, \ldots, v_{t-1}) = p(v_t|v_{t-L}, \ldots, v_{t-1})$$

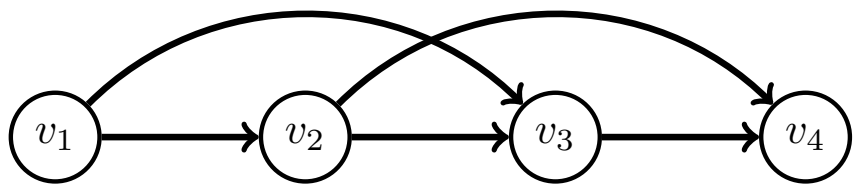
where $L \geq 1$ is the order of the Markov chain

$$p(v_{1:T}) = p(v_1)p(v_2|v_1)p(v_3|v_2)\ldots p(v_T|v_{T-1})$$

For a stationary Markov chain the transitions $p(v_t = s'|v_{t-1} = s) = f(s', s)$ are time-independent ('homogeneous'). Otherwise the chain is non-stationary ('inhomogeneous').



(e)



(f)

Figure : **(a)**: First order Markov chain. **(b)**: Second order Markov chain.

Fitting Markov models

Single series

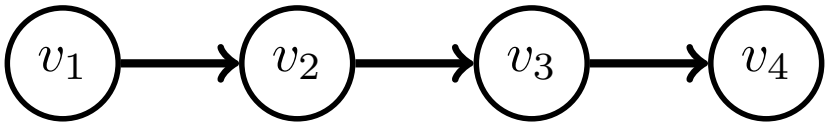
Fitting a first-order stationary Markov chain by Maximum Likelihood corresponds to setting the transitions by counting the number of observed transitions in the sequence:

$$p(v_\tau = i | v_{\tau-1} = j) \propto \sum_{t=2}^{T} \mathbb{I}\left[v_t = i, v_{t-1} = j\right]$$

Multiple series

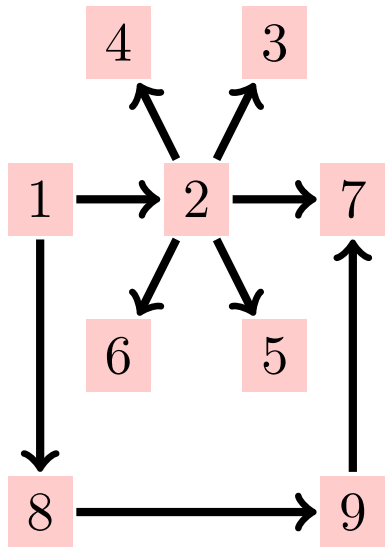
For a set of timeseries, $v^n_{1:T_n}, n = 1, \ldots, N$, the transition is given by counting all transitions across time and datapoints. The Maximum Likelihood setting for the initial first timestep distribution is $p(v_1 = i) \propto \sum_n \mathbb{I}\left[v^n_1 = i\right]$.

Markov Chains

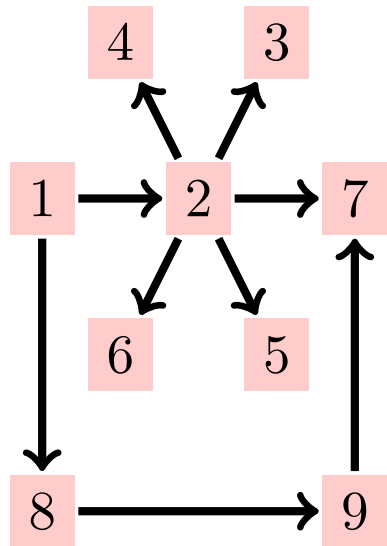


$$p(v_1, \ldots, v_T) = \underbrace{p(v_1)}_{\text{initial}} \prod_{t=2}^{T} \underbrace{p(v_t|v_{t-1})}_{\text{Transition}}$$

'Marginal' inference can be carried out in $O(T)$.
Can use a state-transition diagram to represent $p(v_t|v_{t-1})$



Most probable and shortest paths



- The shortest (unweighted) path from state 1 to state 7 is $1 - 2 - 7$.
- The most probable path from state 1 to state 7 is $1 - 8 - 9 - 7$ (assuming uniform transition probabilities). The latter path is longer but more probable since for the path $1 - 2 - 7$, the probability of exiting state 2 into state 7 is $1/5$.

Equilibrium distribution

It is interesting to know how the marginal $p(x_t)$ evolves through time:

$$p(x_t = i) = \sum_j \underbrace{p(x_t = i|x_{t-1} = j)}_{M_{ij}} p(x_{t-1} = j)$$

The marginal $p(x_t = i)$ has the interpretation of the frequency that we visit state i at time t, given we started from $p(x_1)$ and randomly drew samples from the transition $p(x_\tau|x_{\tau-1})$. As we repeatedly sample a new state from the chain, the distribution at time t, for an initial distribution $\mathbf{p}_1(i)$ is

$$\mathbf{p}_t = \mathbf{M}^{t-1}\mathbf{p}_1$$

If, for $t \to \infty$, $\mathbf{p}_\infty$ is independent of the initial distribution $\mathbf{p}_1$, then $\mathbf{p}_\infty$ is called the equilibrium distribution of the chain.

$$p_\infty(i) = \sum_j p(x_t = i|x_{t-1} = j)p_\infty(j)$$

In matrix notation this can be written as the vector equation

$$\mathbf{p}_\infty = \mathbf{M}\mathbf{p}_\infty$$

so that the stationary distribution is proportional to the eigenvector with unit eigenvalue of the transition matrix.

PageRank

Define the matrix

$$A_{ij} = \begin{cases} 1 & \text{if website } j \text{ has a hyperlink to website } i \\ 0 & \text{otherwise} \end{cases}$$

From this we can define a Markov transition matrix with elements

$$M_{ij} = \frac{A_{ij}}{\sum_{i'} A_{i'j}}$$

- If we jump from website to website, the equilibrium distribution component $p_\infty(i)$ is the relative number of times we will visit website i. This has a natural interpretation as the 'importance' of website i.
- For each website i a list of words associated with that website is collected. After doing this for all websites, one can make an 'inverse' list of which websites contain word w. When a user searches for word w, the list of websites that contain word is then returned, ranked according to the importance of the site.

Hidden Markov Models

The HMM defines a Markov chain on hidden (or 'latent') variables $h_{1:T}$. The observed (or 'visible') variables are dependent on the hidden variables through an emission $p(v_t|h_t)$. This defines a joint distribution

$$p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1)\prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1})$$

For a stationary HMM the transition $p(h_t|h_{t-1})$ and emission $p(v_t|h_t)$ distributions are constant through time.

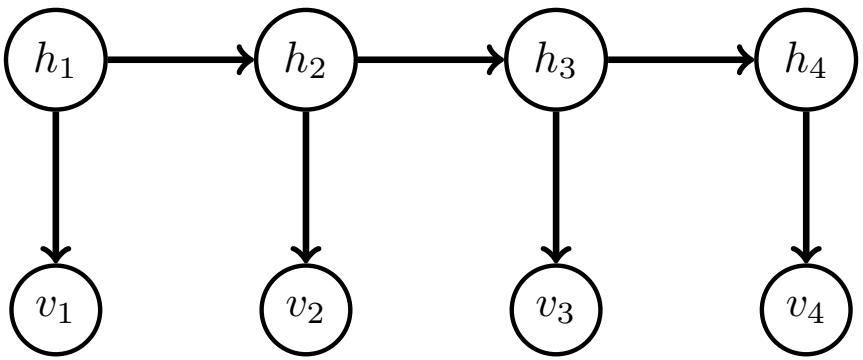


Figure : A first order hidden Markov model with 'hidden' variables $\mathrm{dom}(h_t) = \{1, \ldots, H\}$, $t = 1 : T$. The 'visible' variables v_t can be either discrete or continuous.

HMM parameters

Transition Distribution

For a stationary HMM the transition distribution $p(h_{t+1}|h_t)$ is defined by the $H \times H$ transition matrix

$$A_{i',i} = p(h_{t+1} = i'|h_t = i)$$

and an initial distribution

$$a_i = p(h_1 = i).$$

Emission Distribution

For a stationary HMM and emission distribution $p(v_t|h_t)$ with discrete states $v_t \in \{1, \ldots, V\}$, we define a $V \times H$ emission matrix

$$B_{i,j} = p(v_t = i|h_t = j)$$

For continuous outputs, h_t selects one of H possible output distributions $p(v_t|h_t)$, $h_t \in \{1, \ldots, H\}$.

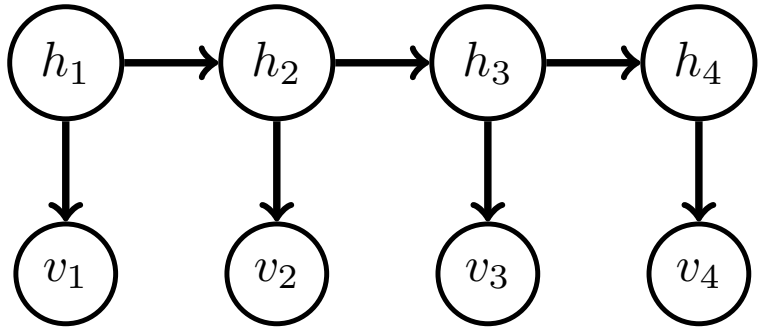
The classical inference problems

Filtering	(Inferring the present)	$p(h_t \mid v_{1:t})$	
Prediction	(Inferring the future)	$p(h_t \mid v_{1:s})$	$t > s$
Smoothing	(Inferring the past)	$p(h_t \mid v_{1:u})$	$t < u$
Likelihood		$p(v_{1:T})$	
Most likely Hidden path	(Viterbi alignment)	$\operatorname{argmax}_{h_{1:T}} p(h_{1:T} \mid v_{1:T})$	

For prediction, one is also often interested in $p(v_t \mid v_{1:s})$ for $t > s$.

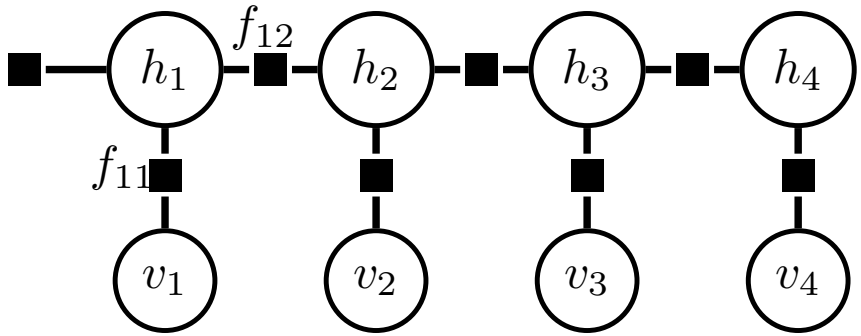
Inference in Hidden Markov Models using Factor Graphs

Belief network representation of a HMM:



As a factor graph:

$f_{11} = p(v_1|h_1)$
$f_{12} = p(h_2|h_1)$



- Filtering: carried our by passing messages up and to the right.
- Smoothing: combine filtering messages with messages up and to the left. Viterbi computed similarly.

Localisation example – Part I

Problem: You're asleep upstairs in your house and awoken by a burglar on the ground floor. You want to figure out where the burglar might be based on a sequence of noise information.

You mentally partition the ground floor into a 5×5 grid. For each grid position

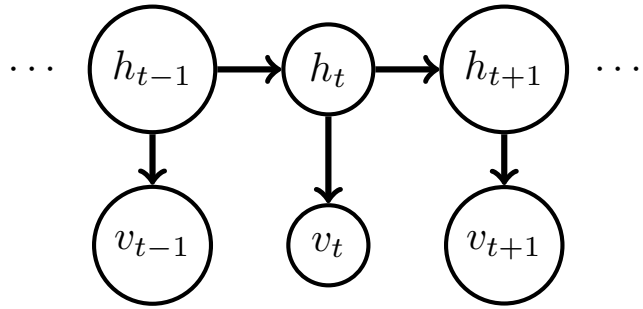
- you know the probability that if someone is in that position the floorboard will creak
- you know the probability that if someone is in that position he will bump into something in the dark
- you assume that the burglar can move only into a neighbor grid square with uniform probability

Prob. of creak

Prob. of bump

Localisation example – Part II

We can represent the scenario using a HMM where



- The hidden variable h_t represents the position of the burglar in the grid at time t

  $h_t \in \{1, \ldots, 25\}$

- The visible variable v_t represents creak/bump at time t

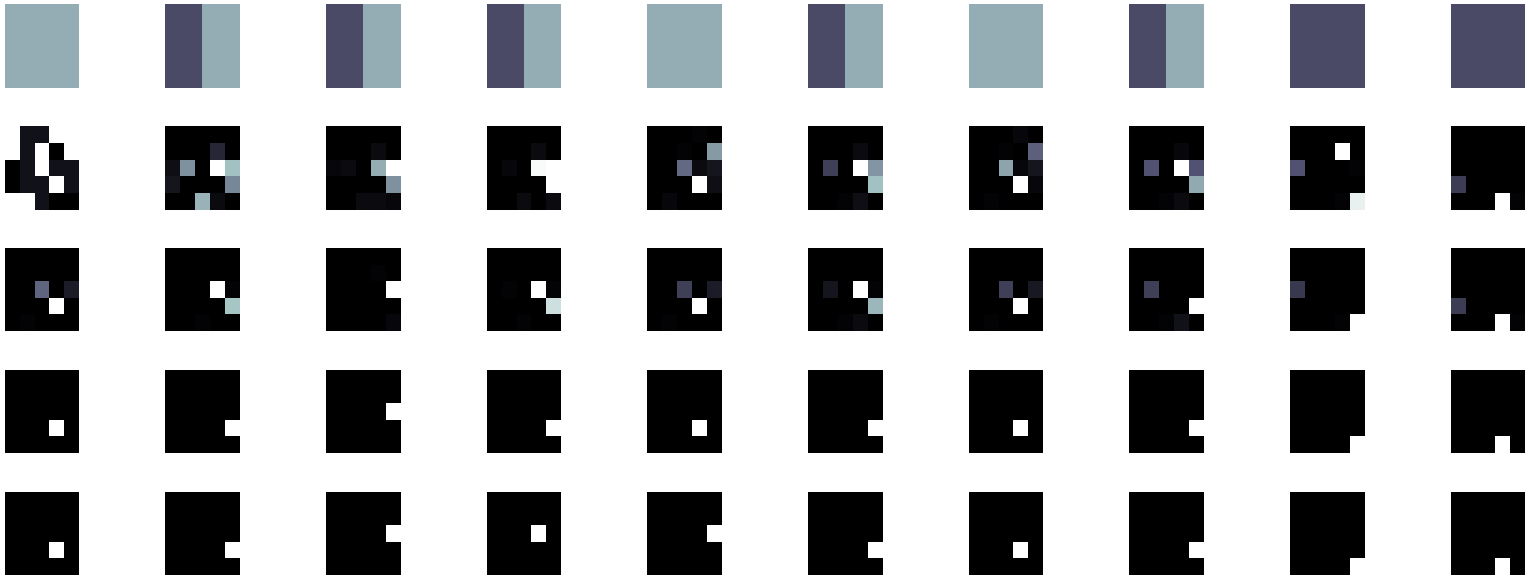
  v=1: no creak, no bump

  v=2: creak, no bump

  v=3: no creak, bump

  v=4: creak, bump

Localisation example – Part III



(top) Observed creaks and bumps for 10 time-steps

(below top) Filtering $p(h_t|v_{1:t})$

(middle) Smoothing $p(h_t|v_{1:10})$

(above bottom) Most likely sequence $\underset{h_{1:T}}{\operatorname{argmax}}\, p(h_{1:T}|v_{1:T})$

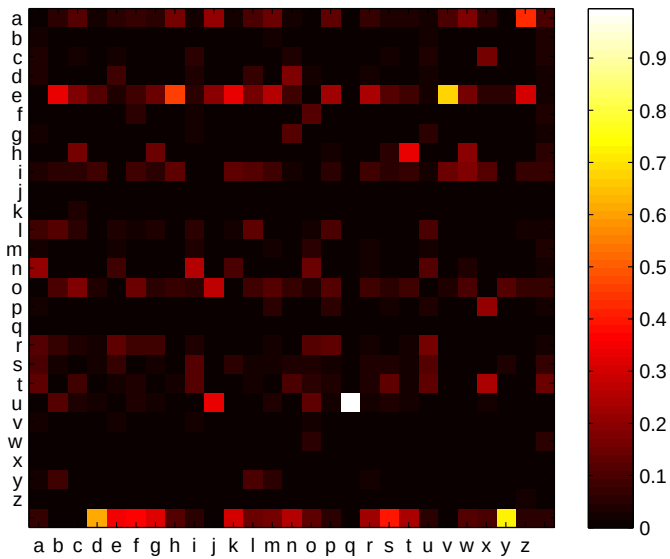
(bottom) True Burglar position

Natural Language Model Example – Part I

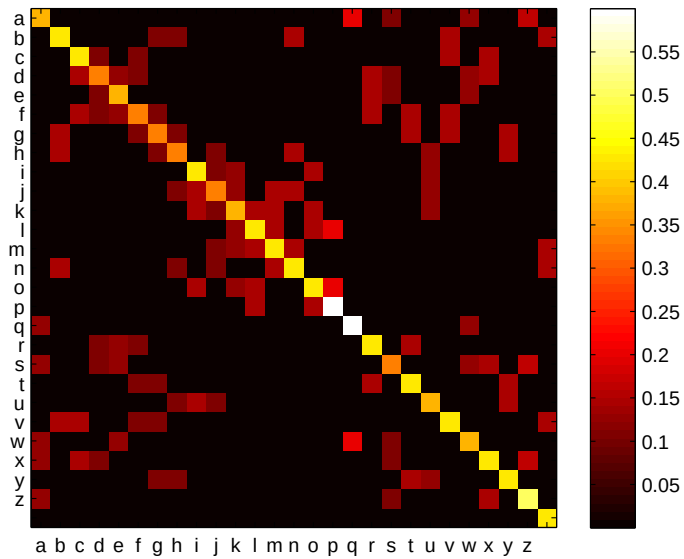
Problem: A 'stubby finger' typist has the tendency to hit either the correct key or a neighbouring key. Given a typed sequence you want to infer what is the most likely word that this corresponds to.

- The hidden variable h_t represents the intended letter at time t
- The visible variable v_t represents the letter that was actually typed at time t

We assume that there are 27 keys: lower case a to lower case z and the space bar.



Transition $p(h_t = j|h_{t-1} = i)$



Emission $p(v_t = j|h_t = i)$

Natural Language Model Example – Part II

Given the typed sequence `kezrninh` what is the most likely word that this corresponds to?

- Listing the 200 most likely hidden sequences (using a form of Viterbi)
- Discard those that are not in a standard English dictionary
- Take the most likely proper English word as the intended typed word

. . . and the answer is . . .

The HMM-GMM

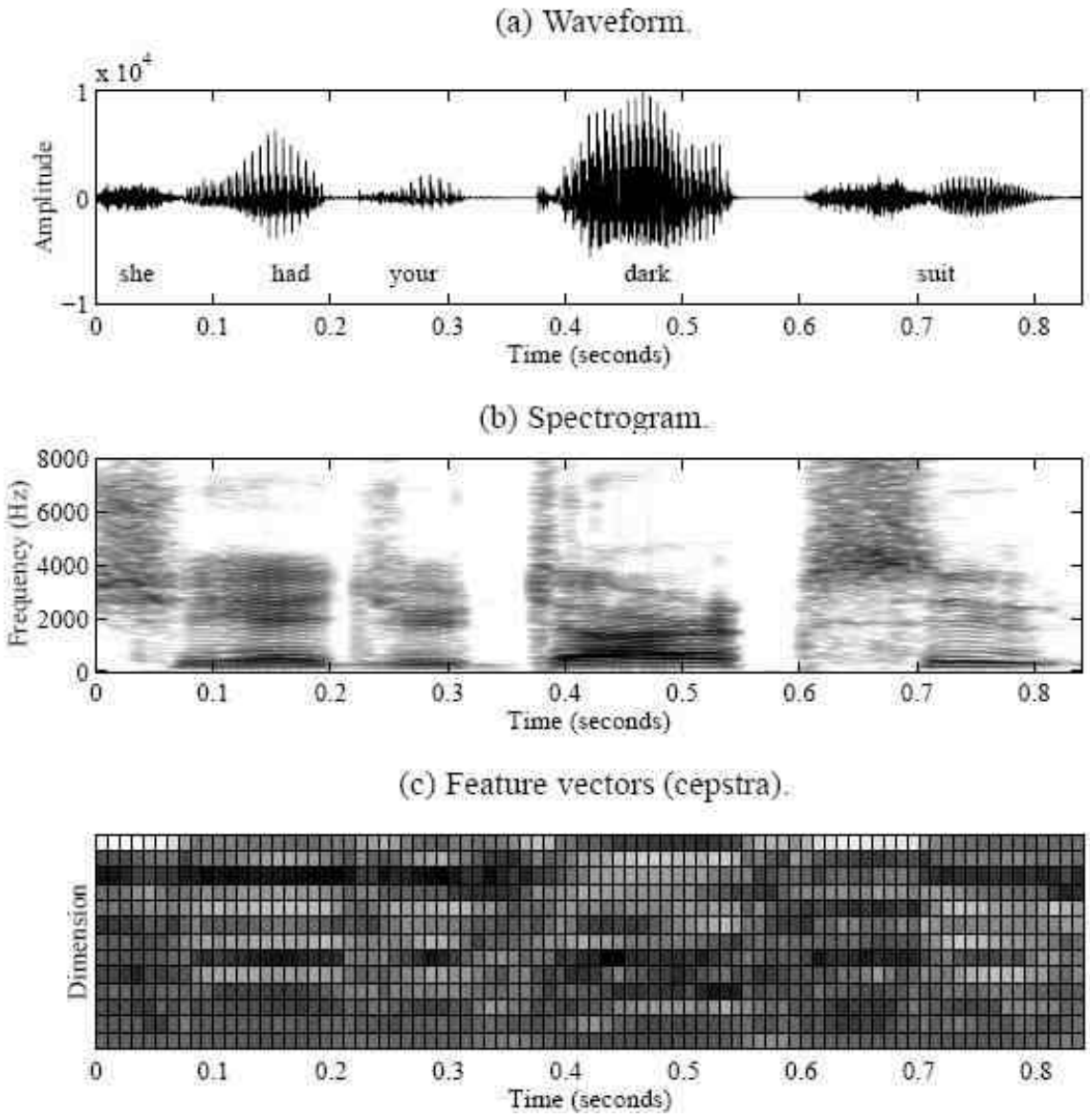
A common continuous observation mixture emission model component is a Gaussian

$$p(\mathbf{v}_t|k_t, h_t) = \mathcal{N}\left(\mathbf{v}_t \middle| \boldsymbol{\mu}_{k_t,h_t}, \boldsymbol{\Sigma}_{k_t,h_t}\right)$$

so that k_t, h_t indexes the $K \times H$ mean vectors and covariance matrices. EM updates for these means and covariances are straightforward. These models are common in tracking applications, in particular in speech recognition (usually under the constraint that the covariances are diagonal).

Message passing inference

Using a continuous output does not change any of the standard inference message passing equations so that inference can be carried out for essentially as before.



h_t is the phoneme at time t. $p(h_t|h_{t-1})$ – language model. $p(v_t|h_t)$ – speech signal model.