Knowledge Representation

COMP2002

Lauren Ansell

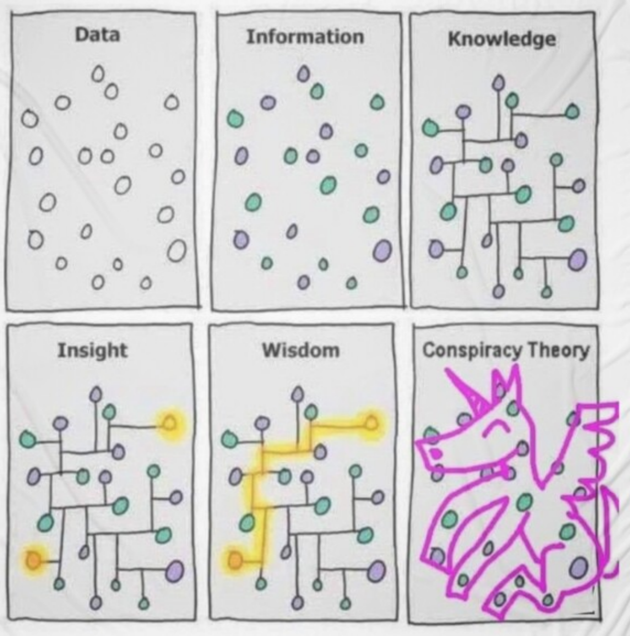
Introduction

Today's topics:

- Logic
- Knowledge representation

Session learning outcomes - by the end of today's lecture you will be able to:

- Use first order predicate logic to represent simple logical statements
- Explain the role of inference within AI
- Identify the components of a knowledge representation scheme


Data
Information
Knowledge
Insight
Wisdom
Conspiracy Theory

Logic-based AI

Introduced by John McCarthy

Base AI on models of logical reasoning rather than simulated cognitive processes.

Proposed a universal problem solving for solving generic problems

Are two people siblings?

Use a **declarative program** to determine if two arbitrary people are siblings – specify **rules** that lead to a solution

```
sibling(X, Y) :- parent(Z, X) and parent(Z, Y)
```

Leave reasoning to the computer

- Specify rules
- Use reasoning to test a hypothesis such as `sibling(John, Mary)?`

Some Knowledge – A Basic Inference

John is married to Mary
John is older than his wife

Mary is not older than her husband

The two premises together allow us to infer the conclusion.

The inference is said to be sound.

An inference $P_1, \ldots, P_n / C$ is said to be valid as long as it is not possible for all of $P_1, \ldots, P_n$ to be true while C is false.

Truth

Conjunction: ⋀

A	B	A ⋀ B
true	true	true
true	false	false
false	true	false
false	false	false

Disjunction: ⋁

A	B	A ⋁ B
true	true	true
true	false	true
false	true	true
false	false	false

Implication – *if A then B…*

$$A \rightarrow B$$

Negation – *not A*

$$\neg A$$

Some Inferences

$$A \wedge B / A$$

If $A \wedge B$ is true then A must be true – hence the conclusion holds.

$$A \vee B / A$$

For $A \vee B$ to be true then either A or B must be true – if B is true then the conclusion doesn't hold

$$A \wedge B / A \vee B$$

If $A \wedge B$ is true then A and B must both be true – this is sufficient for $A \vee B$ to hold.

$$\neg A \vee B, B / A$$

Either $\neg A$ or B must be true – since we also know B is true A might be, but we don't know for sure.

First Order Predicate Logic

Use FOPL to specify rules that describe the world using constants, variables, functions, predicate symbols (e.g., $>, \leq$).

Predicates

- Propositions (e.g., A and B on the previous slides) are replaced with predicates
- A predicate represents a property or relation between terms that can be true or false
- $F(x)$ – x is female, x is Friday...
- $S(x, y)$ – x is y's sibling, x studies y at university...

$\exists$: The Existential Quantifier

John is a poet
Mary is a computer scientist
John loves Mary

Some poet loves some computer scientist

Write logical schemas to represent these premises (binding John to a and Mary to b)

- $Poet(a)$ – meaning a is a poet
- $ComputerScientist(b)$ – meaning b is a computer scientist
- $Loves(a, b)$ – meaning a loves b

Could write this as

For some x, x is a poet and for some y, y is a computer scientist, and x loves y.

Write in FOPL using the existential quantifier $\exists$

$$\exists x(Poet(x) \wedge \exists y(ComputerScientist(y) \wedge Loves(x, y)))$$

$\forall$: The Universal Quantifier

Consider the following statements:

Every integer has a square	For every x, if x is an integer, then x has a square
All people are mortal	For every x, if x is a person then x is mortal
John has read every Shakespeare play	For every x, if x is a Shakespeare play, then John has read it
Each house has a roof	For every x, if x is a house, then x has a roof

Given suitable interpretations of predicates $P(x)$ and $Q(x)$ we can rewrite all of those as follows:

$$\forall x (P(x) \rightarrow Q(x)) \tag{1}$$

Examples

Given predicates:

$Lecture(x)$	x is a lecture
$Student(x)$	x is a student
$Attended(x, y)$	x attended y

No student attended every lecture

$$\neg\exists x(Student(x) \wedge \forall y(Lecture(y) \rightarrow Attended(x, y)))$$

No lecture was attended by every student

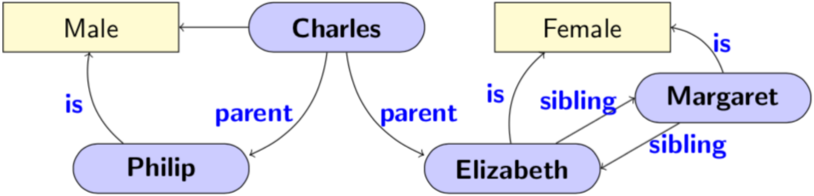
$$\neg\exists x(Lecture(x) \wedge \forall y(Student(y) \rightarrow Attended(y, x)))$$

No lecture was attended by any student

$$\neg\exists x(Lecture(x) \wedge \exists y(Student(y) \wedge Attended(y, x)))$$

Knowledge is encoded as a set of related notions:

- Atomic concept – base idea or class
- Object – an instance of a concept
- Role – describes how two objects relate



Frames

An extension of semantic frameworks – proposed by Minsky in 1970

Replace nodes with complex structures called frames.
A frame consists of slots – used to precisely define properties and features of an object.
Each slot contains facets.

Example factes:

- Value – the current value of the slot stored in this facet
- Range – the range of values allowed (or a list of allowed values)
- Default – the default value of the slot
- Inheritance rules

Scripts

Events follow a standard sequence – e.g. a visit to the doctor

1. Enter the clinic
2. Go to the reception desk to check in
3. Wait your turn until the doctor is free
4. Examination by the doctor
5. Doctor issues a prescription

Define generalized patterns of events – proposed by Schank and Abelson in 1977.

Provides a structural representation of an event.
Useful for reasoning about the course of events – e.g., NLP guesses with the of a script

Script Elements

Agents – objects that can affect other objects (e.g., the doctor, the patient. . .)

Props – things that occur in a script (e.g., a thermometer (to take the patient's temperature), the prescription. . .)

Actions – elementary actions within the script (e.g. entering the clinic, writing the prescription)

Preconditions – propositions that must be true at the start of the script (e.g., patient is ill, clinic is open. . .)

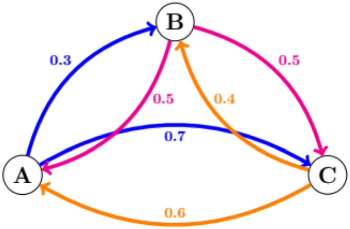
Results – propositions that are true at the moment the script ends (e.g. prescription is written out by a doctor)

Can also group actions into scenes

Graph Theory

$$G = (V, E, W)$$

V is the set of vertices (nodes)
E is the set of edges (directed or undirected)
W is the adjacency matrix with W_{ij} the weight of the edge between nodes i and j



V = {A, B, C}

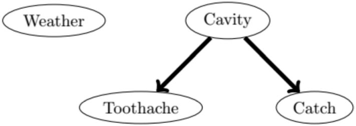
$$W = \begin{bmatrix} 0 & 0.3 & 0.7 \\ 0.5 & 0 & 0.5 \\ 0.4 & 0.4 & 0 \end{bmatrix}$$

Bayesian Networks

Bayesian networks are used to represent independence and conditional independence relationships.

Formally:

- A directed graph in which each node is annotated with quantitative probability information
- Each node corresponds to a random variable
- A set of edges relate variables – if there is an edge from A to B then A is B's parent
- Each node X_i has a conditional probability distribution $P(X_i|Parents(X_i))$ that quantifies the effect of the parents on a node



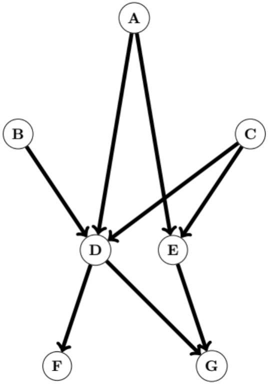
Conditional Independence

An independent variable is not dependent on any of the other variables –

$$P(A) \tag{2}$$

A dependent variable is dependent on the probability of another variable occurring

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{3}$$



If two variables A and B are conditionally dependent given a third C we gain no additional information about A from B if we already know C

$$P(A|B, C) = P(A|C) \tag{4}$$

The full joint distribution of the network is therefore written as

$$P(X_1, ..., X_N) = \prod_1 P(X_i|parents(X_i)) \tag{5}$$

Constructing A Bayesian Network

Nodes

Determine the set of variables that are needed to model the domain
Order them – generally best so that causes precede effects

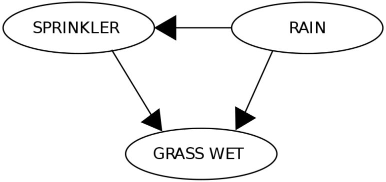
Links

For each node choose a minimal set of parents such that $P(X_i | Parents(X_i))$ is satisfied
Insert a link between each node and each of its parents
Write down the conditional probability table

Bayesian Network - Example

We want to model the dependencies between three variables: a sprinkler, the presence or absence of rain and whether the grass is wet or not.

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

$$\Pr(G, S, R) = \Pr(G \mid S, R) \Pr(S \mid R) \Pr(R)$$

The model can answer questions about the presence of a cause given the presence of an effect.

Question - What is the probability that it is raining, given the grass is wet?

$$\Pr(R = T \mid G = T) = \frac{\Pr(G = T, R = T)}{\Pr(G = T)} = \frac{\sum_{x \in \{T,F\}} \Pr(G = T, S = x, R = T)}{\sum_{x,y \in \{T,F\}} \Pr(G = T, S = x, R = y)}$$

Using the expansion for the joint probability function $\Pr(G, S, R)$ and the conditional probabilities from the conditional probability tables stated in the diagram, one can evaluate each term in the sums in the numerator and denominator.

For example,

$$\begin{aligned}
\Pr(G = T, S = T, R = T) &= \Pr(G = T \mid S = T, R = T)\,\Pr(S = T \mid R = T)\,\Pr(R = T) \\
&= 0.99 \times 0.01 \times 0.2 \\
&= 0.00198
\end{aligned}$$

Then the numerical results are

$$\begin{aligned}
\Pr(R = T \mid G = T) &= \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} \\
&= \frac{891}{2491} \approx 35.77\%.
\end{aligned}$$

Decision Graphs

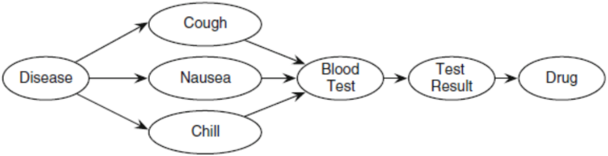
Modified Bayesian network – don't want to simply consider random variables

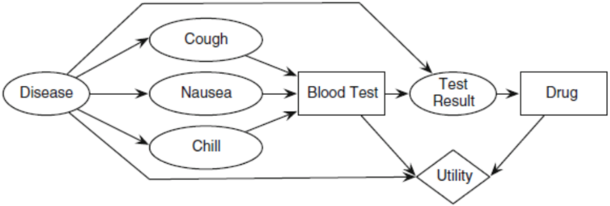
Model decision points

Decisions are shown in temporal order

Utilities (costs) are modelled based on decisions and random nodes

Decision Graphs - Example





Blood Test and Drug are decision points – utility measures medical and financial cost of the sequence of the two decisions

Summary

Logic-based AI

- Propositional calculus
- First order predicate calculus
- Inference

Knowledge representation

- Ontologies and semantic networks
- Frames and scripts
- Bayesian Networks