Lecture #10: Reasoning Under Uncertainty and Vagueness - Part 1

COMP3608: Intelligent Systems
Inzamam Rahaman

Thinking rationally

- For most of the course, we looked at rational agents
  - Emphasised approaches used to build systems that behave rationally
- For the remainder of the course, we will look at techniques for building systems that “think” rationally

Thinking Rationally

- What does it mean to think rationally?
- First term that ought to come to mind is logic
- What is a logic?

The Nature of Logic

- What is Logic?
  - Prepositions: a statement that can be true or false
    - Example: the sky is blue
    - Counterexample: eat your veggies
      - Subtle distinction here between “eat your veggies” and “you should each your veggies”!
  - Informally: A logic is calculus that defines connectives used to build formulae from simple prepositions and also define a notion of inference using those connectives

Types of Logic

- There are many different types of logic developed by philosophers, mathematicians, and computer scientists over several millennia
- Each with different connectives, used in different contexts with different goals, and different assumptions
- Examples: propositional logic, predicate logic, modal logic, fuzzy logic, intuitionistic logic, probabilistic reasoning, etc…

The Role of Logic in Building Intelligent Systems

- Logic is used directly in AI in to build expert systems, build decision support systems, and reason about outcomes and causes

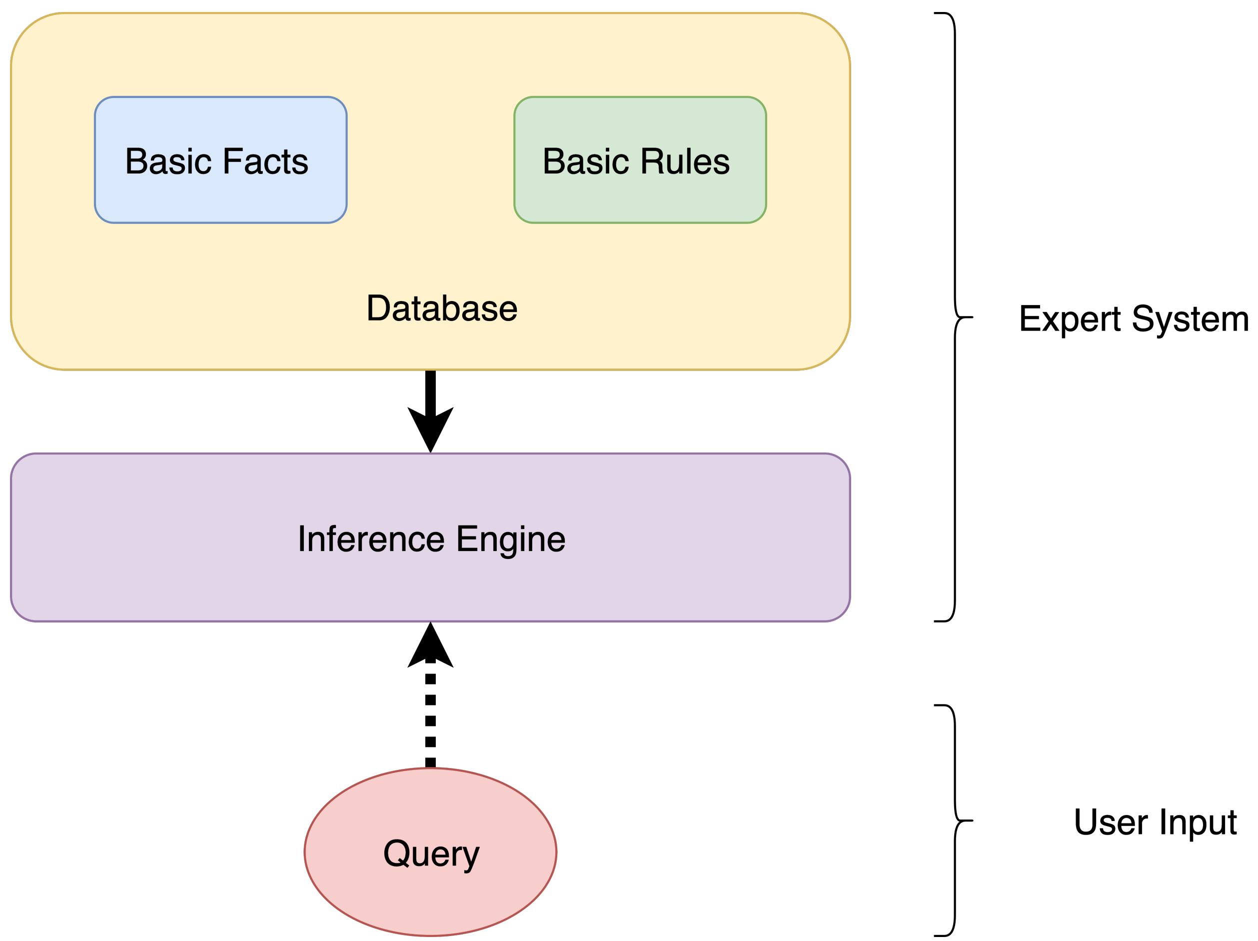
  - We use logic to facilitate the answering of queries (things we want to know)

- The field of AI dedicated to leveraging logic is called KRR - Knowledge Reasoning and Representation

- How do we effectively represent knowledge about a domain and then use those representations to answer queries (i.e. reason)

Expert Systems

- Logical formulae capture relationships
- Idea: have expert help us encode relationships from a domain as logical expressions
  - Called knowledge engineering
  - Generates a database of facts and rules
- Have inference engine that enables the derivation of new facts from basic facts and rules
  - Inference engine operates in accord with a particular logic
- Expose interface for a user to submit or make queries to find out something
  - User wants facts



Rules and Facts

- Expert Systems require effort to design initial structure of rules and facts
- Some KRR techniques such as Bayesian Networks admit some degree of machine learning
- Techniques such as Inductive Logic Programming and Genetic Algorithm admit learning some structure
- But higher up-front development costs for large expert systems
- But also more interpretable
    - Conclusions of the system come from traceable logic rather matrix multiplication by seemingly magical matrices

Predicate Logic

- Will review predicate logic briefly
  - Also called First-order logic (FOL)
- Predicate logic uses same connectives from propositional logic
- Adds titular predicates - abstracts relationships over domain using functions
- Add quantifiers - for all $\forall$ and there exists $\exists$ that better qualify the domain for which the proposition is true
- Reasoning by hand is done using instantiation procedures
- Basis of simple expert systems

Predicate Logic Example

- Let
  - $H(x)$ be x is human
  - $M(x)$ be x is mortal
- Given the fact $H(\text{Socrates})$
- Given the rule $\forall x(H(x) \implies M(x))$
- User queries $M(\text{Socrates})$

```
human(socrates).

mortal(X) :- human(X).

?- mortal(socrates).
```

Automated Reasoning with FOL

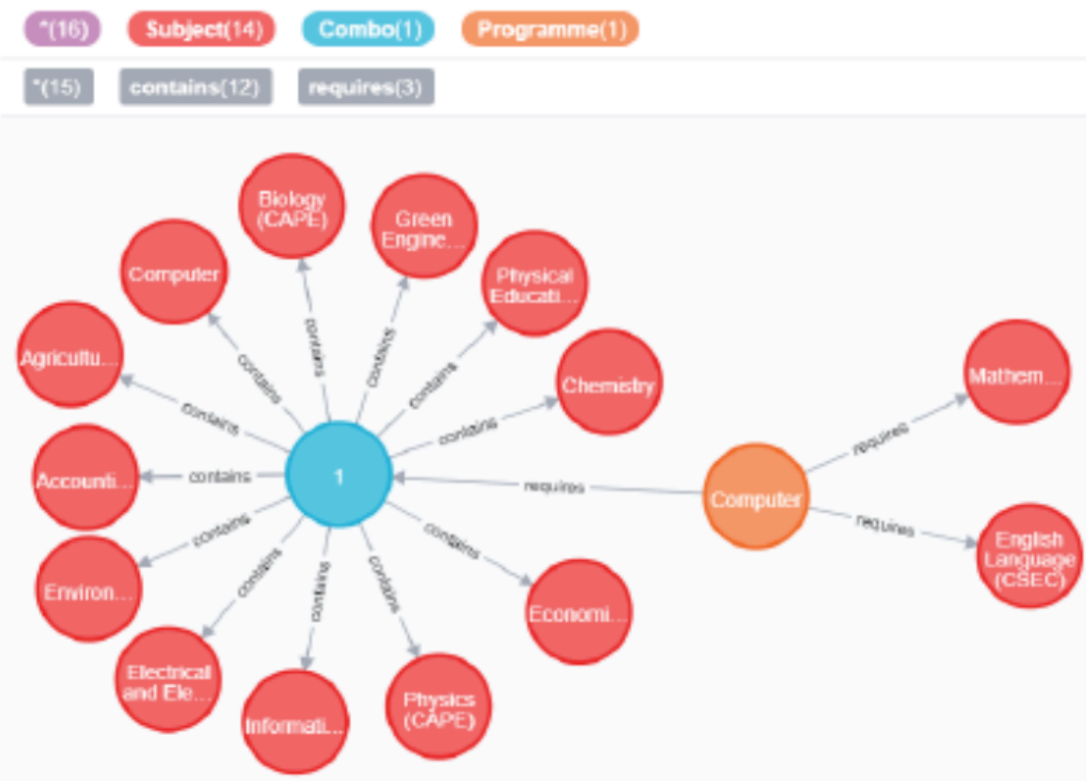
- Programming languages such as Prolog use techniques such as unification to help answer queries
- Problem of satisfying query can be seen as a constraint-satisfaction problem
- Uses backtracking-based algorithm called unification
- Other systems more tailored for expert system development, such as CLIPS, use other mechanisms for tracking which rules can “fire”

Graphical Representations

- Sometimes we can capture logical relationships between entities in our domain using graphs
- Nodes represent entities
- Edges represent relationships
  - Edges have relationship type property
- Using these Knowledge graphs help us in terms of efficiency
  - Make querying answering and fact derivation a graph traversal problem

Mendez et al. 2019

- Specialised case of a system where we have a finite set of logical rules (that can be arbitrarily expanded)
- Want to find all logical formulae that are satisfied by user input
- Model knowledge-base as a graph that encodes logical rules
- Used in BeUWI App



Predicate Logic Problems

- Predicate logic is not very expressive when it comes to
  - Uncertainty - relationships in the real-world are better described in terms of probabilities
  - Vagueness - given that Tom’s height and a relationship that determines if some is *Tall* from their height, can we infer that Tom is *Very Tall*?
- Need alternatives for many complex scenarios!

Probability to the Rescue

- The whole point of probability is to allow us a way to reasoning when confronted with uncertainty

- In particular, we will look at how Bayes Theorem and related ideas in probability allow us robust ways to reason

- Querying in this context amounts to asking about the probability of some random variable taking on a certain value or querying about the distribution of some random variable

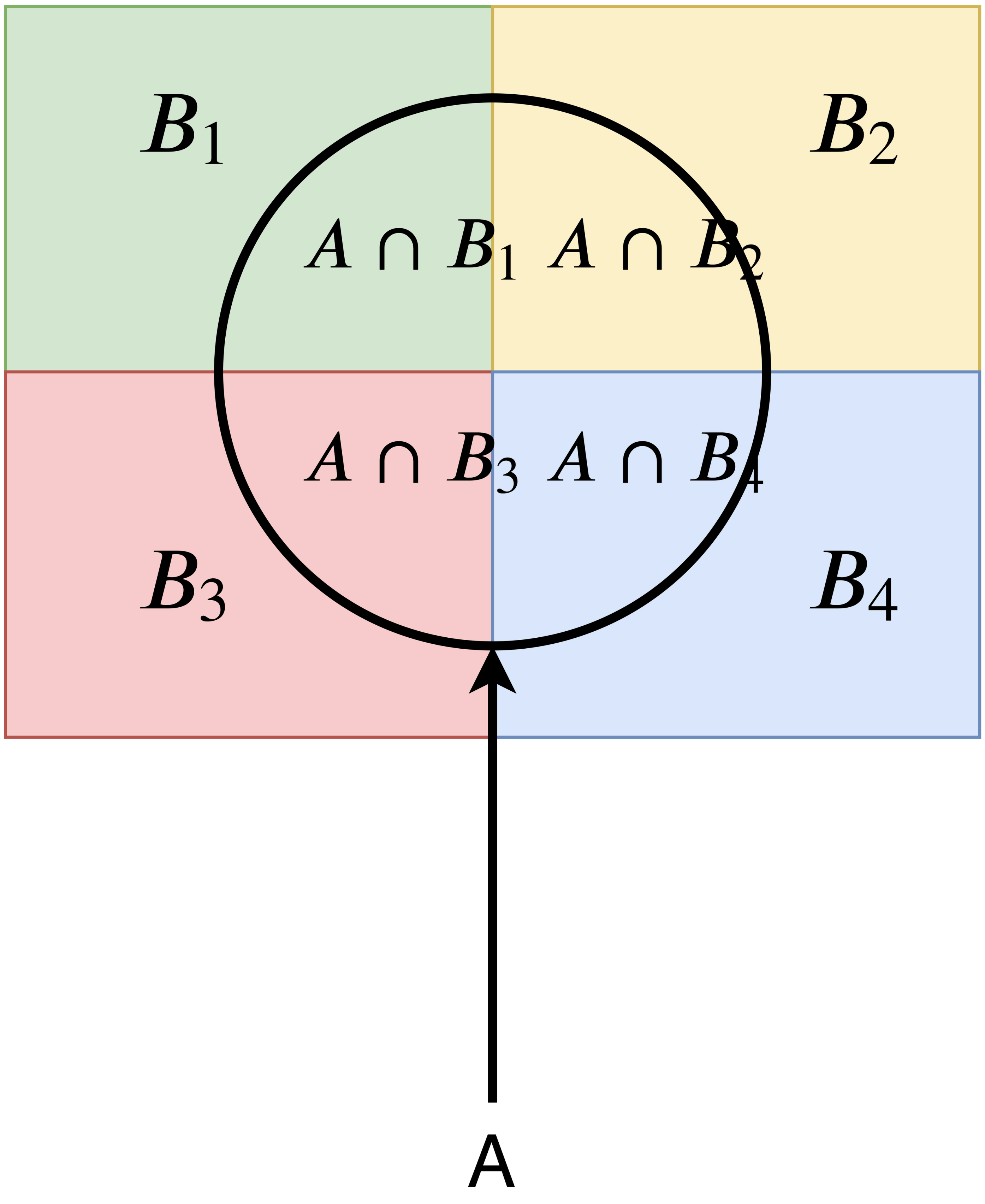
- Note: everything we discuss here is generalisable to continuous random variables, but we will assume discrete random variables going forward

Probability Recaps

- Law of Total Probability. Suppose that we have events $A, B_1, B_2, \ldots B_n$ where $\forall i, j$ where $i \neq j : B_i \cap B_j = \varnothing$ then

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i)$$

Can also write $P(X \cap Y)$ as $P(X, Y)$



B_1
B_2
$A \cap B_1$
$A \cap B_2$
$A \cap B_3$
$A \cap B_4$
B_3
B_4
A

Bayes Theorem

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)}$$

$$P(X \mid Y)P(Y) = P(X \cap Y)$$

$$P(Y \mid X)P(X) = P(Y \cap X) = P(X \cap Y)$$

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Query

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Evidence

Bayes Theorem Example

- Suppose that A is the event that you have the flu and B is the event that you coughed.

- Assuming that $P(A) = 0.05$ and $P(B\,|\,A) = 0.80$, calculate $P(A\,|\,B)$

Normalisation

- Suppose that we want the distribution of $P(X \mid Y)$
- Recall that from Bayes rule $P(X \mid Y) = \dfrac{P(X, Y)}{P(Y)}$
- Hence to compute $P(X \mid Y = y)$ where y is a particular value for random variable Y, we need would need to consider all possible values for random variable X
- We can make our computation shorter by using normalisation!

Normalisation

- The function of the denominator in Bayes theorem is to normalise the values from the numerator, i.e. make the probabilities add up to 1
- So we just need to compute the numerator across all possible values of X and then normalise them

Normalisation - Example

C \ T	Toothache	No Toothache
Cavity	0.12	0.1
No Cavity	0.08	0.7

Normalisation - Example

- Compute distribution of $P(C \mid T = \text{No Tootache})$

$$P(C \mid T = \text{No Tootache}) =$$

$$\alpha < P(C = \text{Cavity}, T = \text{No Tootache}), P(C = \text{No Cavity}, T = \text{No Tootache}) >$$

$$\alpha < 0.1{,}0.7 >$$

$$P(C \mid T = \text{No Tootache}) = < 0.13{,}0.87 >$$

Normalisation

- Normalisation is sometimes more efficient than computing the denominator
- Sometimes, in very complicated cases, the denominator is intractable
    - Cannot easily compute it!

Chain Rule of Probability

$$P(X_1, X_2, \ldots X_n) = P(X_n \mid X_{n-1}, \ldots X_1) P(X_{n-1}, \ldots X_1)$$

$$P(X_1, X_2, \ldots X_n) = P(X_n \mid X_{n-1}, \ldots X_1) P(X_{n-1} \mid X_{n-2} \ldots X_1) P(X_{n-2}, \ldots X_1)$$

$$P(X_1, X_2, \ldots X_n) = \prod_{i=1}^{n} P\left(X_i \mid \bigcap_{j=1}^{i-1} X_j \right)$$

Chain Rule of Probability

$$P(X_1, X_2, \ldots X_n) = P(X_n \,|\, X_{n-1}, \ldots X_1) P(X_{n-1}, \ldots X_1)$$

$$P(X_1, X_2, \ldots X_n) = P(X_n \,|\, X_{n-1}, \ldots X_1) P(X_{n-1} \,|\, X_{n-2} \ldots X_1) P(X_{n-2}, \ldots X_1)$$

$$P(X_1, X_2, \ldots X_n) = \prod_{i=1}^{n} P\left(X_i \,|\, \bigcap_{j=1}^{i-1} X_j \right)$$

Core insight: we can factor a joint probability in terms of several conditional probabilities!

Reasoning Probabilistically - Joint Probabilities

- Suppose that we have several of random variables $X_1, X_2, \ldots X_n$
- We have a joint probability distribution over these variables
- This joint probability distribution is a powerful tool!

Reasoning using Joint Probabilities

- Suppose that we have access to $P(X_1, X_2, \ldots X_n)$, and that we have observations for several of the random variables
- Suppose that we observed values for several random variables. Let's call this configuration our evidence
- We can use our evidence to compute the probabilities the different values of a query variable
  - Can get the distribution of the values of the query variable
  - We want to compute $P(\text{Query Variable} \mid \text{Evidence})$

Example

	Toothache		No Toothache	
	Catch	No Catch	Catch	No Catch
Cavity	0.108	0.012	0.072	0.008
No Cavity	0.016	0.064	0.144	0.576

Example

- Suppose we know that our patient has a toothache
- Want to query if they have a cavity or not
- Don't know if the probe hit the patient's teeth too hard
- We want to compute $P(C \mid T = \text{toothache})$
- Let's see how we can do this using the joint probabilities

Go To whiteboard

Problems

- We have a bit of a problem
- Consider that we have two settings for each random variable
- How many probabilities do we need to store if we have n random variables?

Problems

- We have a bit of a problem
- Consider that we have two settings for each random variable
- How many probabilities do we need to store if we have n random variables?
  - 2^n entries in a table
  - This is intractable for non-trivial problems
  - Need a more efficient representation

Conditional Independence

- Not all random variable “affect” one another
- Think about casual relationships between random variables
- My dog barking has nothing to do if Alice has a toothache
- Can use the notion of conditional independence to reduce quantity of data we need

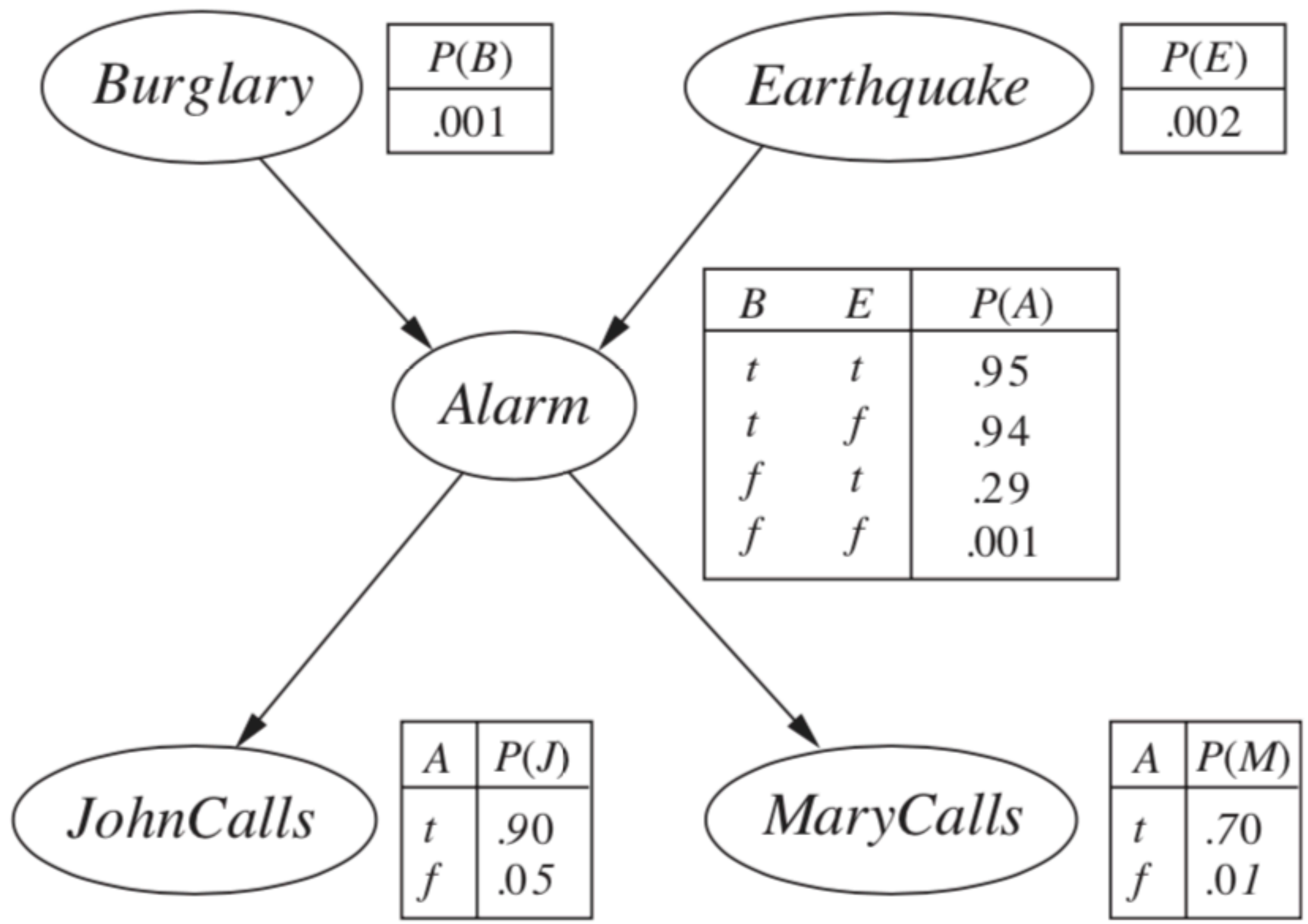
Conditional Independence

- We say that X is conditionally independent of Y given Z iff $P(X, Y | Z) = P(X | Z)P(Y | Z)$
  - We can more succinctly say that $(X \perp Y) | Z$
  - Some textbooks omit the brackets
- Recall the chain rule
- If we know that some random variables are conditionally independent of one another, we can “factor” these relationships out

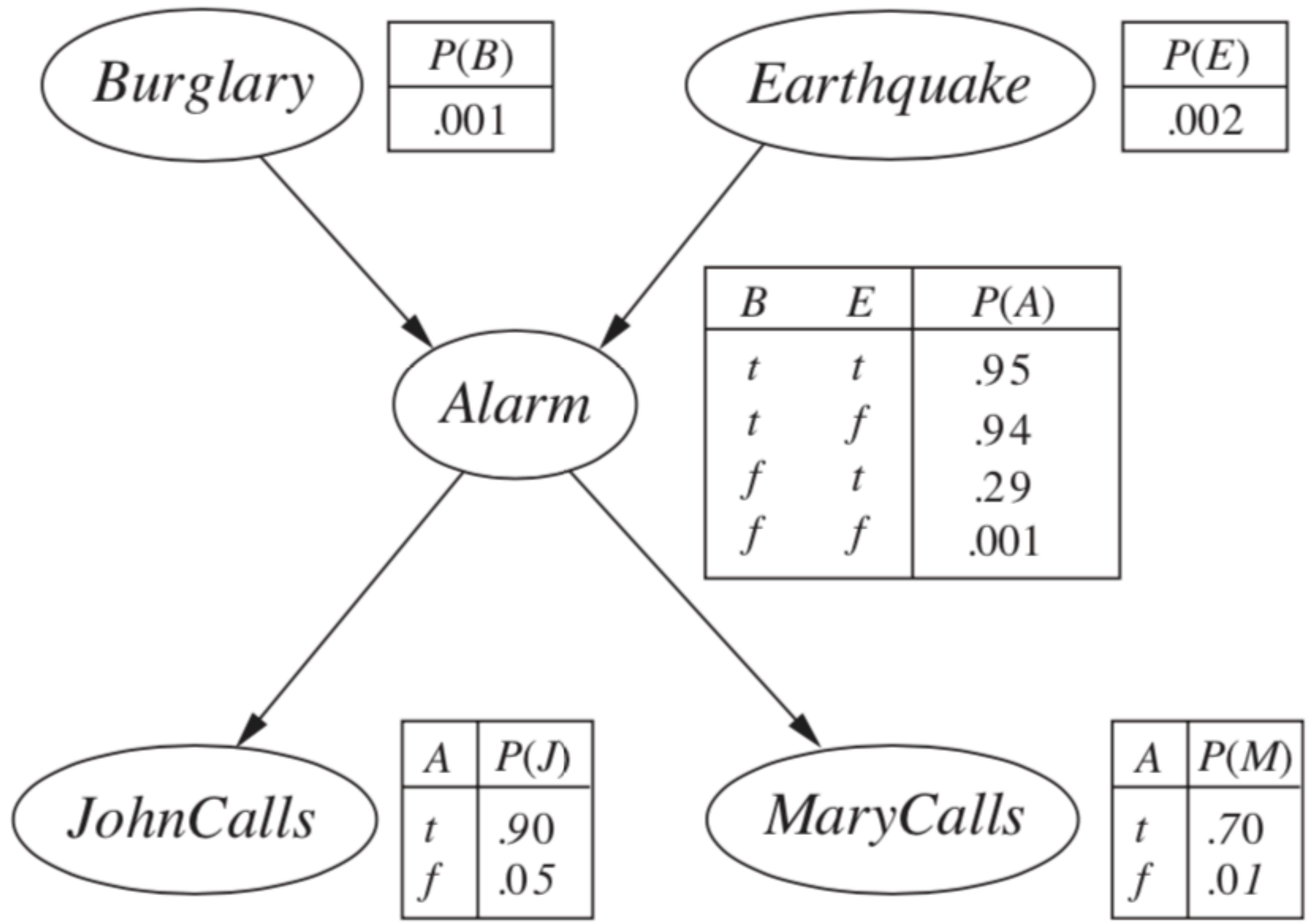
Bayesian Networks

Bayesian Networks

- A graphical way to represent a joint probability distribution while capturing conditional independence relationships
- Powerful knowledge representation tool
- Also useful for casual relationship modelling
- Judea Pearl - picture adjacent - won Turing Prize for their invention and subsequent use in casual modelling
  - Revolutionised many other areas of science


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

$$P(X_1, X_2, \ldots X_n) = \prod_{i=1}^{n} P(X_i \,|\, \mathrm{Parents}(X_i))$$



$$P(B, E, A, J, M) = P(B)P(E)P(A \mid B, E)P(J \mid A)P(M \mid A)$$

Example

Consider that both Mary and John call us. Was there a Burglary?

Bayesian Networks

- Each node has a property called its Markov Blanket
- A node's Markov Blanket is the set of its children, parents, and its co-parents (children's parents)
- Example, Markov Blanket of Alarm (A) is $\{$Buglary, Earthquake$\}$
- What about JohnCalls?

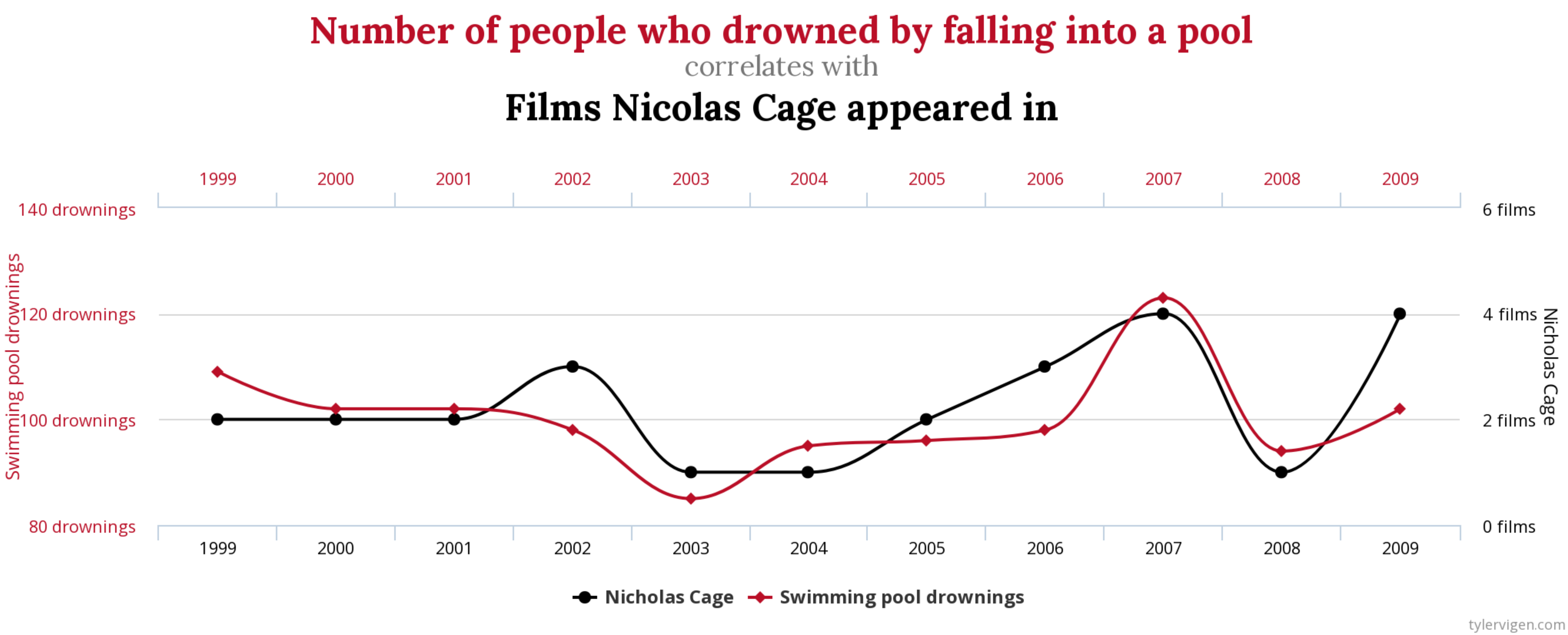
Bayesian Networks

- In a Bayesian Network we say that a node's probability distribution is conditionally independent of all other nodes given its markov blanket

Learning CPDs in Nodes

- We can use machine learning techniques to help us compute the probability distributions in the nodes
  - Use EM algorithm
- Also use methods to learn edges given nodes
  - Still open question on how to do well

Causality



Number of people who drowned by falling into a pool
correlates with
Films Nicolas Cage appeared in
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
140 drownings
120 drownings
100 drownings
80 drownings
Swimming pool drownings
6 films
4 films
2 films
0 films
Nicholas Cage
Nicholas Cage
Swimming pool drownings
tylervigen.com

Causal Reasoning

- One of the most amazing things about human beings is our ability to work in terms of causal models
- We can pose counterfactual queries to our mental models
  - Ask “What if?”
  - Neural Networks as cool as they are cannot answer “What if?” questions very well…

Causality

- Remember that one of the most interesting things about bayesian networks is that we can represent casual relations
- Pearl's do-calculus allows us rigorous way to reason in terms of interventions
  - What if we do X?
- Will look at do-calculus next week and some intractability problems next week
- Also do fuzzy logic and wrap up main course content