CmpE 540
Principles of Artificial Intelligence

Pınar Yolum
pinar.yolum@boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

Uncertainty

(Based mostly on the course slides from
http://aima.cs.berkeley.edu/ and
http://www.cmpe.boun.edu.tr/~akin/)

Uncertainty

- Our knowledge is incomplete and changing to include new facts
- How can we solve problems in the face of this incompleteness?
- Logic fails in such cases - but humans are capable of solving problems in these situations
- We need methods that can handle uncertainty

Sources of Uncertainty

- Incorrectness
  – Human error
  – Equipment error
  – False negative - rejecting a true hypothesis
  – False positive - accepting a false hypothesis
- Measurement
  – Precision, how precise e.g. cm or mm
  – Accuracy - calibration

Sources of Uncertainty

- Random
  - Due to noise, loose wires etc.
- Systematic
  - Due to error in reading
    - e.g. user giving inches not cm
- Reasoning
  - Wrongly formulated or learnt rules

5

Example of reasoning with uncertainty

- Abbott, Babbitt and Cabot are suspects in a murder case
- *Abbott's alibi:* registration in a respectable hotel in Albany
- *Babbitt's alibi:* was visiting his brother-in-law in Brooklyn at the time of the murder
- *Cabot's alibi:* watching a ski meet in the Catskills (but no witnesses)

6

Example of reasoning with uncertainty (Cont'd)

- We have the following beliefs:
  - Abbott, Babbitt, or Cabot committed the murder
  - Abbott did not commit the murder (alibi)
  - Babbitt did not commit the murder (alibi)
  - Cabot may have committed the murder (his alibi is weak)
- We must solve this problem based on "beliefs" using "belief states"
  - I believe that Cabot is the murderer because he has the weakest alibi

7

Example of reasoning with uncertainty (Cont'd)

- Now consider that Cabot is able to document his alibi (someone took a photograph of him watching the ski meet)
  - Cabot's alibi is now stronger
- Babbitt's brother-in-law might be lying
  - Babbitt's alibi is now weaker
- Abbott's registry could be forged
  - Abbott's alibi is now weaker
- How do we deal with such uncertainties? What conclusion(s) should we draw?

8

Symbolic Approaches

Monotonic Reasoning

- Conventional reasoning mechanisms, such as logic, assume:
  - Knowledge used is 'complete', i.e. all facts and rules required are in the knowledge base.
  - Facts that are necessary to solve the problem are present or can be derived from the knowledge provided in the knowledge base.
  - Facts and knowledge are consistent, i.e. new facts or knowledge does not invalidate old ones.
  - Facts are either known to be true or false
  - Inferences can be established as either true or false.
- Such a reasoning mechanism is called *monotonic reasoning*
- If any of the above properties is not satisfied, conventional logic-based reasoning mechanisms become inadequate.

Nonmonotonic Reasoning

- Extending the knowledge base
  - Make inferences when information not available
  - Differentiate between: *it is known that a fact is not true* and *the fact is not known*
- Updating information in the knowledge base
  - When a new fact is added
  - Infer new facts from these
- Resolving conflicts when several inconsistent inferences are made.
  - Assign preferences or
  - Allow for multi-solution inferences

Nonmonotonic Logic

- A system where statements can be T, F or neither (knowledge does not have to be complete)
- There is no formalism like Resolution, so we must develop more "ad hoc" methods.
- Add an operator M to first order predicate calculus
- M means "True if consistent"

Nonmonotonic Logic

- Now we are able to prove things T and F with statements that include M because it means that we assume any future information will be consistent. If not, our proof does not hold
- Example:
  - $\forall$X, Y: related (X,Y) $\wedge$M Get Along (X,Y) $\Rightarrow$Will Defend (X, Y)
  - That is, we assume that if X and Y are related then X will defend Y. However, if we learn that X does not get along with Y then this becomes false.

13

Default Logic

- A logic where assumptions are made when rules are introduced

A : B ---> "If A is provable and it is
C consistent to assume B then
conclude C"

- This is different from Non-monotonic logic by requiring the assumption to be made at the time a rule is introduced (i.e. you cannot add new rules of this form)

14

Abduction

- Standard deduction (*Modus Ponens*) states that if you know A is true and you have the rule $A \Rightarrow B$, then you can conclude B is true. This is truth preserving
- In Abduction, we turn this around to read if B is true and you have a rule A $\Rightarrow$ B then you can conclude A is true. This is not truth preserving.

15

Abduction

- Abduction is more often used when we have several rules A1 $\Rightarrow$ B, A2 $\Rightarrow$ B, A3 $\Rightarrow$ B, A4 $\Rightarrow$ B, etc...
- If B is true, what caused it (what was responsible for B being true)? Our answer is A1 or A2 or A3 or A4, but which one?
- We can use our beliefs to pick the right one.
- Abduction is often used for explanations:
  - diagnosis/credit assignment
  - theory understanding/legal reasoning
  - recognition/perception/natural language understanding

16

Reconsider the murder case

- We assume Cabot's alibi is good because of the photograph.
- We assume that Abbott really went to the hotel because of the hotel's reputation.
- We believe that Babbitt's brother might lie (he has been known to lie before)
- Therefore, we conclude that Babbitt is the murderer

17

Implementations

- Augmenting a problem solver with "UNLESS"
- Dependency-Directed Backtracking
- Justification-based Truth Maintenance System
- Assumption-based Truth Maintenance System

18

Using UNLESS

- Augment rules with the *Unless* clause which introduces exceptions to a rule
- Use *forward chaining* if the problem is data-driven
- Use *backward chaining* if the problem is goal-driven - this approach will not be able to determine if/when new information becomes available and thus is not as useful

19

Dependency-Directed Backtracking

- Consider the following example:
  - We want to prove F
  - We have a nonmonotonic rule If $A \Rightarrow F$
  - We have no reason not to assume A, so we assume A is true
  - $F \Rightarrow G \wedge H$
  - By some other facts, we can also derive $M \wedge N$
  - We learn A is not true. Backtracking would remove F, G, H, $M \wedge N$. F, $G \wedge H$ may not be true, but $M \wedge N$ were derived independently!

20

Truth Maintenance Systems

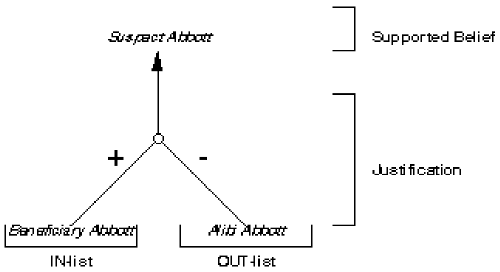
- A Truth Maintenance System (TMS) is responsible for:
  - Enforcing logical relations among beliefs.
  - Generating explanations for conclusions.
  - Finding solutions to search problems
  - Supporting default reasoning.
  - Identifying causes for failure and recover from inconsistencies.
  - The TMS / IE relationship is the following:

Families of TMSs

- There are several families of TMSs, which differ in the representation scheme they use and the functionality they support:
  - **Justification-based TMSs.** The language used is limited to Horn formulas.
  - **Logic-based TMSs.** These use a full propositional logic language.
  - **Assumption-based TMSs.** Language limited to Horn formulas, but several alternatives (contexts) can be explored at the same time.
  - **Non-monotonic JTMSs.** Language limited to Horn formulas, but allow non-monotonic justifications, thus making it possible to implement default reasoning.
  - **Clause Management Systems.** Their representational power is equivalent to LTMSs, but like ATMSs can support several contexts at the same time.
  - **Contradiction-tolerant TMSs.** Language limited to Horn formulas, but support non-monotonic and plausible reasoning and deal explicitly with contradictions in a single context.

Justification-Based TMS

- To justify (believe) an assertion, all items in the "IN-list" must be believed while all items in the "OUT-list" must be disbelieved.



A breadth-first approach

- JTMS - depth-first search of possible beliefs with dependency-directed backtracking
- Here, backtracking is avoided by considering multiple belief states at one time
- As the search space is expanded, any states that are inconsistent are pruned away reducing some of the search

Numerical Approaches

Probabilistic Reasoning

Statistical Reasoning

- Nonmonotonic reasoning can be used to model belief systems in which facts are believed to be true, false or unknown.
- However, for some problems, it is useful to be able to describe beliefs that are not certain but for which there is some supporting evidence.

Classes of Problems

- There are two broad classes of problems.
  - Genuine randomness of the world.
    - Playing cards or throwing dice are examples of such problems.
    - Model statistically
    - Predict likelihood of various outcomes, though complete certainty is not possible.
  - World not random, but we don't know the state.
    - Many 'common sense tasks' fall into this class, such as, in certain domains, fault diagnostic and engineering design.
    - Difficult to model and enumerate exceptions
    - Better to summarize statistically

Probability

Probabilistic assertions summarize effects of
- laziness: failure to enumerate exceptions, qualifications, etc.
- ignorance: lack of relevant facts, initial conditions, etc.

Subjective probability:
- Probabilities relate propositions to agent's own state of knowledge
  - e.g., $P(A_{25}$ | no reported accidents$) = 0.06$

These are not assertions about the world

Probabilities of propositions change with new evidence:
- e.g., $P(A_{25}$ | no reported accidents, 5 a.m.$) = 0.15$

29

Making decisions under uncertainty

Suppose I believe the following:

$P(A_{25}$ gets me there on time | …) = 0.04
$P(A_{90}$ gets me there on time | …) = 0.70
$P(A_{120}$ gets me there on time | …) = 0.95
$P(A_{1440}$ gets me there on time | …) = 0.9999

- Which action to choose?

  Depends on my preferences for missing flight vs. time spent waiting, etc.
  - Utility theory is used to represent and infer preferences
  - Decision theory = probability theory + utility theory

30

Syntax

- Basic element: random variable
  - Similar to propositional logic: possible worlds defined by assignment of values to random variables.
- Boolean random variables
  - e.g., *Cavity* (do I have a cavity?)
- Discrete random variables
  - e.g., *Weather* is one of <*sunny,rainy,cloudy,snow*>
- Domain values must be exhaustive and mutually exclusive
- Elementary proposition constructed by assignment of a value to a random variable: e.g., *Weather = sunny*, *Cavity = false* (abbreviated as $\neg$*cavity*)
- Complex propositions formed from elementary propositions and standard logical connectives e.g., *Weather = sunny* $\vee$ *Cavity = false*

31

Syntax

- Atomic event: A complete specification of the state of the world about which the agent is uncertain

  E.g., if the world consists of only two Boolean variables *Cavity* and *Toothache*, then there are 4 distinct atomic events:

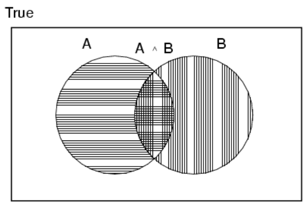
  Cavity = false $\wedge$ *Toothache = false*
  Cavity = false $\wedge$ *Toothache = true*
  Cavity = true $\wedge$ *Toothache = false*
  Cavity = true $\wedge$ *Toothache = true*

- Atomic events are mutually exclusive and exhaustive

32

Axioms of probability

- For any propositions *A*, *B*
  - $0 \leq P(A) \leq 1$
  - P(*true*) = 1 and P(*false*) = 0
  - $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Prior probability

- Prior or unconditional probabilities of propositions
  e.g., P(*Cavity* = true) = 0.1 and P(*Weather* = sunny) = 0.72 correspond to belief prior to arrival of any (new) evidence
- Probability distribution gives values for all possible assignments:
  P(*Weather*) = <0.72,0.1,0.08,0.1> (normalized, i.e., sums to 1)
- Joint probability distribution for a set of random variables gives the probability of every atomic event on those random variables
  P(*Weather,Cavity*) = a 4 × 2 matrix of values:

Weather =	sunny	rainy	cloudy	snow
Cavity = true	0.144	0.02	0.016	0.02
Cavity = false	0.576	0.08	0.064	0.08

- Every question about a domain can be answered by the joint distribution

Conditional probability

- Conditional or posterior probabilities
  e.g., P(*cavity* | *toothache*) = 0.8
  i.e., given that *toothache* is all I know
- Notation for conditional distributions:
  P(*Cavity* | *Toothache*) = 2-element vector of 2-element vectors
- If we know more, e.g., *cavity* is also given, then we have
  P(*cavity* | *toothache,cavity*) = 1
- New evidence may be irrelevant, allowing simplification, e.g.,
  P(*cavity* | *toothache, sunny*) = P(*cavity* | *toothache*) = 0.8
- This kind of inference, sanctioned by domain knowledge, is crucial

Conditional probability

- Definition of conditional probability:
  $P(a \mid b) = P(a \wedge b) / P(b)$ if $P(b) > 0$
- Product rule gives an alternative formulation:
  $P(a \wedge b) = P(a \mid b) P(b) = P(b \mid a) P(a)$
- A general version holds for whole distributions, e.g.,
  P(*Weather,Cavity*) = **P**(*Weather* | *Cavity*) **P**(*Cavity*)
- (View as a set of 4 × 2 equations, not matrix mult.)
- Chain rule is derived by successive application of product rule:

$$\mathbf{P}(X_1, \ldots, X_n) = \mathbf{P}(X_1,\ldots,X_{n-1})\, \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1})$$
$$= \mathbf{P}(X_1,\ldots,X_{n-2})\, \mathbf{P}(X_{n-1} \mid X_1,\ldots,X_{n-2})\, \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1})$$
$$= \ldots$$
$$= \pi_{i=1}^{n} \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$$

Inference by enumeration

- Start with the joint probability distribution:

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- For any proposition φ, sum the atomic events where it is true: $P(\varphi) = \Sigma_{\omega:\omega \models \varphi} P(\omega)$

Inference by enumeration

- Start with the joint probability distribution:

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- For any proposition φ, sum the atomic events where it is true: $P(\varphi) = \Sigma_{\omega:\omega \models \varphi} P(\omega)$
- P(*toothache*) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2

Inference by enumeration

- Start with the joint probability distribution:

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- For any proposition φ, sum the atomic events where it is true: $P(\varphi) = \Sigma_{\omega:\omega \models \varphi} P(\omega)$
- P(cavity v *toothache*) = 0.108 + 0.012 + 0.016 + 0.064 + 0.72 + 0.08 = 0.28

Inference by enumeration

- Start with the joint probability distribution:

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- Can also compute conditional probabilities:

$$P(\neg cavity \mid toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)} = \frac{0.016+0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

Normalization

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

- Denominator can be viewed as a normalization constant α

P(*Cavity* | *toothache*) = α, **P**(*Cavity,toothache*)
= α, [**P**(*Cavity,toothache,catch*) + **P**(*Cavity,toothache,¬ catch*)]
= α, [<0.108,0.016> + <0.012,0.064>]
= α, <0.12,0.08> = <0.6,0.4>

General idea: compute distribution on query variable by fixing evidence variables and summing over hidden variables

Inference by enumeration

Typically, we are interested in
the posterior joint distribution of the query variables **Y**
given specific values **e** for the evidence variables **E**

Let the hidden variables be **H** = **X** - **Y** - **E**

Then the required summation of joint entries is done by summing out the hidden variables:
$\mathbf{P}(\mathbf{Y} \mid \mathbf{E} = \mathbf{e}) = \alpha\mathbf{P}(\mathbf{Y},\mathbf{E} = \mathbf{e}) = \alpha\Sigma_h\mathbf{P}(\mathbf{Y},\mathbf{E}= \mathbf{e}, \mathbf{H} = \mathbf{h})$

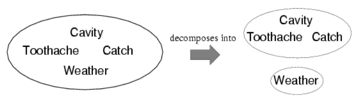
- The terms in the summation are joint entries because **Y**, **E** and **H** together exhaust the set of random variables
- Obvious problems:
  1. Worst-case time complexity $O(d^n)$ where d is the largest arity
  2. Space complexity $O(d^n)$ to store the joint distribution
  3. How to find the numbers for $O(d^n)$ entries?

Independence

- **A is (unconditionally) independent of B** if P(A|B) = P(A). In this case, P(A ∧ B) = P(A)P(B).
- **A is conditionally independent of B given C** if P(A|B ∧ C) = P(A|C) and, symmetrically, P(B|A ∧ C) = P(B|C).
  - What this means is that if we know P(A|C), we also know P(A|B ∧ C), so we don't need to store this case. Furthermore, it also means that
  - P(A ∧ B|C) = P(A|C)P(B|C).

Independence

- *A* and *B* are independent iff
P(*A*|*B*) = **P**(*A*) or **P**(*B*|*A*) = **P**(*B*) or **P**(A, B) = **P**(*A*) **P**(*B*)



P(*Toothache, Catch, Cavity, Weather*)
= **P**(*Toothache, Catch, Cavity*) **P**(*Weather*)

- 32 entries reduced to 12; for *n* independent biased coins, $O(2^n) \to O(n)$
- Absolute independence powerful but rare
- Dentistry is a large field with hundreds of variables, none of which are independent. What to do?

Conditional independence

- **P**(*Toothache, Cavity, Catch*) has $2^3 - 1 = 7$ independent entries

- If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache:
  (1) **P**(*catch* | *toothache, cavity*) = **P**(*catch* | *cavity*)

- The same independence holds if I haven't got a cavity:
  (2) **P**(*catch* | *toothache*,¬*cavity*) = **P**(*catch* | ¬*cavity*)

- *Catch* is conditionally independent of *Toothache* given *Cavity*:
  P(*Catch* | *Toothache,Cavity*) = **P**(*Catch* | *Cavity*)

- Equivalent statements:
  P(*Toothache* | *Catch, Cavity*) = **P**(*Toothache* | *Cavity*)
  P(*Toothache, Catch* | *Cavity*) = **P**(*Toothache* | *Cavity*) **P**(*Catch* | *Cavity*)

Conditional independence

- Write out full joint distribution using chain rule:
  P(*Toothache, Catch, Cavity*)
  = **P**(*Toothache* | *Catch, Cavity*) **P**(*Catch, Cavity*)
  = **P**(*Toothache* | *Catch, Cavity*) **P**(*Catch* | *Cavity*) **P**(*Cavity*)
  = **P**(*Toothache* | *Cavity*) **P**(*Catch* | *Cavity*) **P**(Cavity)
  I.e., 2 + 2 + 1 = 5 independent numbers

- In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in *n* to linear in *n*.

- Conditional independence is our most basic and robust form of knowledge about uncertain environments.

Bayes' Rule

- Product rule P(a∧b) = P(a | b) P(b) = P(b | a) P(a)
  ⇒ Bayes' rule: P(a | b) = P(b | a) P(a) / P(b)
- or in distribution form
  P(Y|X) = **P**(X|Y) **P**(Y) / **P**(X) = α**P**(X|Y) **P**(Y)

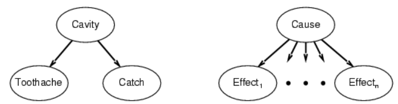
- Useful for assessing diagnostic probability from causal probability:
  – P(Cause|Effect) = P(Effect|Cause) P(Cause) / P(Effect)

  – E.g., let *M* be meningitis, *S* be stiff neck:
    P(m|s) = P(s|m) P(m) / P(s) = 0.8 × 0.0001 / 0.1 = 0.0008
  – Note: posterior probability of meningitis still very small!

Bayes' Rule and Conditional Independence

P(*Cavity* | *toothache* ∧ *catch*)
= α**P**(*toothache* ∧ *catch* | *Cavity*) **P**(*Cavity*)
= α**P**(*toothache* | *Cavity*) **P**(*catch* | *Cavity*) **P**(*Cavity*)

- This is an example of a naïve Bayes model:
  P(Cause,Effect_1, … ,Effect_n) = **P**(Cause) π_i**P**(Effect_i|Cause)



- Total number of parameters is linear in *n*

Certainty Factors

Mycin

- Mycin is the oldest and best known expert system
- Its function is to diagnose certain bacterial infections and recommend a drug treatment

Mycin

- Rules are of the form:

  IF: 1) The strain of the organism is gramneg

  AND

  2) The morphology of the rod is coccus

  THEN:

  There is strongly suggestive evidence (0.8) that the genus of the organism is neisseria

Handling uncertainty: Certainty Factors

- Everything in Mycin has a certainty factor associated with it
- CFs range from –1..1
  - -1 means definitely not
  - +1 means definitely
  - 0 means equally likely and unlikely

Certainty Factors

- User can input CFs in the range –10..10, which are converted to the range –1..1
- These CFs are propagated through the rules
- CFs in the range –0.2..0.2 are not propagated as their values are too small

53

Propagating CFs

- For a rule of the form:
  - If A and B and C then D
- The CF associated with D is found by taking the minimum CF of A, B and C and then multiplying it by the CF associated with the rule

54

Propagating CFs

- For a rule of the form:
  - If A or B or C then D
- The maximum CF associated with A, B or C is taken and then multiplied by the CF associated with the rule

55

Propagating CFs

- If two CFs are derived for the same conclusion they are combined as follows:
- New CF = CF1 + CF2 – (CF1)(CF2)
  - If both CF1 & CF2 > 0
- New CF = CF1 + CF2 + (CF1)(CF2)
  - If both CF1 & CF2 < 0
- New CF = (CF1 + CF2) / (1-min(|CF1|,|CF2|))
  - Otherwise

56

Coments on Certainty Factors

- These calculations are not probabilistic
- They assume that the conclusions are independent
- These problems are a restriction on the viability of Mycin, however as long as the same conclusion isn't reached too often this shouldn't create a problem
- Mycin has a very shallow search space

57

Examples

- If the following are known:
  - A CF(0.4)
  - B CF(0.7)
  - C CF(0.9)
- And the rule
  - If A and B and C then E(0.4)

58

Examples

- For ands we take min
- So min of 0.4, 0.7, 0.9 = 0.4
- Multiply this by the CF for the rule (0.4)
  - Gives 0.16
- This is < 0.2 and so is ignored

59

Examples

- Given:
  - A CF(0.4)
  - B CF(0.7)
  - C CF(0.9)
- And the rule:
  - If A or B or C then E(0.4)

60

Examples

- For or's we take max
- So max of 0.4, 0.7, 0.9 gives 0.9.
- Multiplying by CF of rule (0.4)
  - Gives 0.36
- This is > 0.2 so the CF for E becomes 0.36

61

Examples

- Given:
  - A CF(0.7)
  - B CF(0.8)
  - C CF(0.5)
  - E CF(0.4)
- And the rule
  - If A or B or C then E (0.7)

62

Examples

- Or's so max of 0.7, 0.8, 0.5 = 0.8
- Multiply this by CF for rule 0.8*0.7 = 0.56
- > 0.2 so new CF for E is 0.56
- Already have a CF for E of 0.4
- Need to combine them

63

Examples

- As both > 0 1st rule applies
  - New CF = CF1 + CF2 – (CF1)(CF2)
    = 0.4 + 0.56 – 0.4 * 0.56
    = 0.96 – 0.224
    = 0.736

  New CF for E is therefore 0.736

  This reinforces existing CF

64

Theoretical Flaws in Mycin

- Mycin assumes that a failure to establish the truth of something, despite attempts to do so means that it is false
- The threshold of 0.2 is arbitrary
- Mycin detects and rejects circularity in rules such as:
  - If A then B
  - If B then C
  - If C then A
  - By ignoring the first rule
  - This is not always the best approach
- Self referencing rules cause a problem
- The ordering of the rules is ∴ important