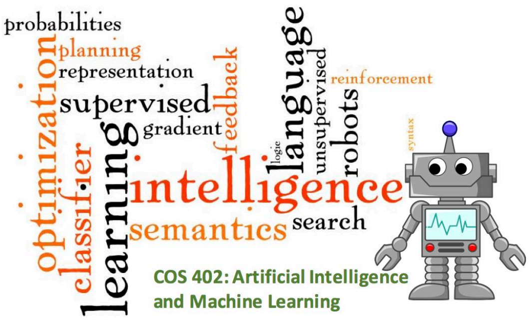


COS 402 – Machine
Learning and
Artificial Intelligence
Fall 2016

Lecture 12: Knowledge Representation and Reasoning Part 1: Logic

Sanjeev Arora Elad Hazan

PRINCETON UNIVERSITY

(Borrows from slides of Percy Liang, Stanford U.)

Thus far in the course

- Formalization of learning from data (statistical learning theory)
- Language models and language semantics. (examples of unsupervised learning)
- Recommender systems.

Today: Knowledge representation and reasoning using logic

Reminder: In-class midterm this Thurs. 75 min; closed book (arrive on time!)
(Study guide posted on piazza.)

LOGIC

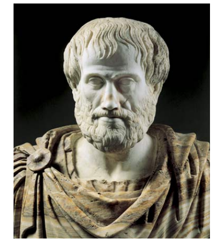
log·ic

/ˈläjik/

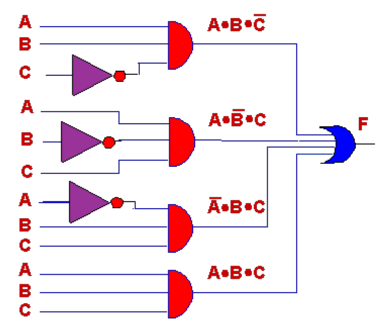
noun

noun: **logic**

1. reasoning conducted or assessed according to strict principles of validity.
   "experience is a better guide to this than deductive logic"
   synonyms: reasoning, line of reasoning, rationale, argument, argumentation
   "the logic of their argument"
   - a particular system or codification of the principles of proof and inference.
     "Aristotelian logic"



Also basis of digital circuits in computer chips
EE206/COS306



Role of logic in AI

- For 2000 years, people tried to codify "human reasoning" and came up with logic.
- Most AI work until 1980s: Build machines that represent knowledge and do reasoning via logic. "Rule based reasoning."
- "Learning from data" is popular today, but lacks aspects that were trivial in the pre-1980s systems (e.g. allow human programmer to easily communicate his/her knowledge to the system). "How do you teach a deep net to multiply two numbers?"
- Logical reasoning now seems poised for a comeback.

Goals of logic

- Represent knowledge about the world.

- Reason with that knowledge.

Natural language?

- A **dime** is better than a **nickel.**
- A **nickel** is better than a **penny.**
- Therefore, a **dime** is better than a **penny.**

Knowledge

Reasoning

- A **penny** is better than a **nothing.**
- **Nothing** is better than **world peace.**
- Therefore, a **penny** is better than **world peace.**

Natural language is tricky!

Use of logic removes ambiguity (similar to computer languages);
but also makes system less flexible. (Will study more flexible versions later.)

Components of any logical system

- Syntax
- Semantics.
- Reasoning

Different syntax, same semantics

$2 + 3 \Leftrightarrow 3 + 2$

Same syntax, different semantics

3/2 in Python 2.7 vs 3/2 in Python 3.

Propositional Logic (aka Boolean Logic; remember COS 126!)

Syntax:

Propositional symbols (atomic formulas): A, B, C

Logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Build up formulas recursively—if f and g are formulas, so are the following:

- Negation: $\neg f$
- Conjunction: $f \wedge g$
- Disjunction: $f \vee g$
- Implication: $f \rightarrow g$
- Biconditional: $f \leftrightarrow g$

$$(A \vee \neg B) \wedge (\neg A \vee B)$$

$\vee$ and $\wedge$ and $\longleftrightarrow$ are symmetric, like “+” and “times” in arithmetic.

Syntax provides symbols.

No “meaning” yet (semantics)!

Semantics provided by a “Model”
(unrelated to “model” used in machine learning!)

For propositional logic, a model is simply an assignment to all variables.
(each variable assigned 0 (false) or 1 (true), not both)

Sanity check: What is # of possible models if there are 3 variables? How about n variables?

Interpretation function

I(f, w): Given formula f and model w, assigns exactly one of 1 (True) or 0 (False) to f.

Build up formulas recursively—if f and g are formulas, so are the following:

- Negation: $\neg f$
- Conjunction: $f \wedge g$
- Disjunction: $f \vee g$
- Implication: $f \rightarrow g$
- Biconditional: $f \leftrightarrow g$

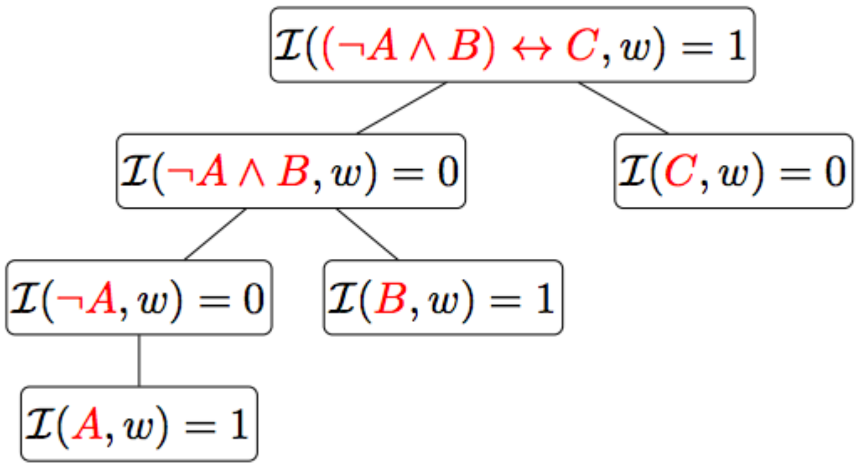
- True iff f is false
- True iff both f and g are true
- True iff at least one of f, g is true
- False iff f is true and g is false.
- True iff f and g have the same value (true or false)

Example: interpretation function

Formula: $f = (\neg A \wedge B) \leftrightarrow C$

Model: $w = \{A : 1, B : 1, C : 0\}$

Interpretation:



Definition:
M(f) = Set of models w for which I(f, w) = True

Formula f compactly represents M(f) ("Set of possible worlds where f is true.")

- Example: For f= A ∧ B

  M(f) = { A =1, B =1}.

- For f = A ↔ B

  M(f) = {A=1, B=1}; {A =0; B =0}

Tautology: Formula f such that M(f) = All possible models. ("True in all possible worlds")

Example: A ∨ ¬A.
(True whether A =0 or A=1!)

Contradiction: Formula f such that M(f) = Empty set. ("False in all possible worlds.")

Example: A ∧ ¬A.
(False whether A =0 or A=1!)

Knowledge representation via logic

Knowledge base : Set of formulae $\{f_1, f_2, \ldots, f_n\}$
M(KB) = All possible models for $f_1 \wedge f_2 \wedge \ldots \wedge f_n$

Formulae = "known facts "
Models = all possible "worlds" where all these facts hold
(Adding more facts to KB can only shrink set of possible worlds.)

Example: Variables: R, S, C ("Rainy", "Sunny," "Cloudy")

KB: $R \vee S \vee C$; ("It is either Rainy or Sunny or Cloudy.")
$R \rightarrow C \wedge \neg S$; ("If it is Rainy then it is Cloudy and not Sunny.")
$C \leftrightarrow \neg S$ ("If it is Cloudy then it is not Sunny, and vice versa")

Models for KB: {R=1, S =0, C =1}; {R =0, C=1, S=0}; {R=0, C=0, S=1}.

Satisfiability

Defn: Knowledge-base KB is satisfiable if

$M(KB) \neq \varnothing$

(i.e. there is some assignment to variables that makes all formulae in KB evaluate to True)

Defn: KB contradicts formula f if

KB $\cup$ {f} is not satisfiable

Defn: KB entails formula f (denoted KB $\models$ f) if

$M(KB \cup \{f\}) = M(KB)$.

(in every world where KB is true, f is also true)

Defn: KB is consistent with formula f if

$M(KB \cup \{f\})$ is non-empty

(there is a world in which KB is true and f is also true)

Sanity check: KB entails f iff it contradicts $\neg f$.

An example

Example: Variables: R, S, C (“Rainy”, “Sunny,” “Cloudy”)

KB: $R \lor S \lor C$;
$R \rightarrow C \land \neg S$;
$C \leftrightarrow \neg S$

Does KB $\vDash S \lor C$?

Models for KB: {R=1, S =0, C =1}; {R =0, C=1, S=0}; {R=0, C=0, S=1}.

$S \lor C$ is true in all these models ✔

An example

Example: Variables: R, S, C (“Rainy”, “Sunny,” “Cloudy”)

KB: $R \lor S \lor C$;
$R \rightarrow C \land \neg S$;
$C \leftrightarrow \neg S$

Add a variable: U (“Carry an umbrella”). What common-sense “facts” can we add about U to the above KB?

Examples: $R \rightarrow U$; $S \rightarrow \neg U$;

AI systems till 1980s used such reasoning; ”facts” were added by programmers based upon introspection.

Decision-making at run-time = which formulae are entailed/contradicted/consistent

Recap of logic so far

- Defn of formulae.
- KB = List of formulae. (“Facts about the world”)
- KB can entail or contradict another formula, or be consistent with it.
- To decide which of the three possibilities of prev. line holds, draw up list of all possible models. (“Truth table method.”)

Truth table method (to check if KB has any model)

- If n variables, can take 2^n time. (infeasible for even n =100)
  Any faster algorithm?

- Polynomial time algorithm ➔ P = NP (Famous open problem)

- In practice there are reasonable algorithms that use resolution and other related reasoning methods.

Resolution procedure to decide satisfiability of a KB

(simplest version; [Davis-Putnam, 1950s])

KB consists only of formulae that are clauses (ie $\vee$ of variables or negated variables).

(With some work, can convert every KB to this form.)

Warmup: What can we conclude under foll. conditions?

KB has singleton clauses (A), ($\neg A$).

HAS NO MODEL (UNSATISFIABLE)!

KB contains clause pairs of form ($A \vee B_1 \vee \ldots \vee B_n$) and ($\neg A \vee C_1 \vee \ldots \vee C_m$)

Every model for KB must make ($B_1 \vee \ldots \vee B_n \vee C_1 \vee \ldots \vee C_m$) TRUE

Resolution procedure to decide satisfiability of a KB

(simplest version; [Davis-Putnam, 1950s])

KB consists only of formulae that are clauses (ie $\vee$ of variables or negated variables).

(With some work, can convert every KB to this form.)

Claim (won't prove): Finishes in finite time for every KB and prints correct answer.

While KB nonempty **do**
{
If KB contains clause pairs of form (A), ($\neg$ A)
    Print ("No Model.") and **STOP.** /*WHY?*/
If KB contains clause pairs of form ($A \vee B_1 \vee \ldots \vee B_n$) and ($\neg A \vee C_1 \vee \ldots \vee C_m$)
    Add ($B_1 \vee \ldots \vee B_n \vee C_1 \vee \ldots \vee C_m$) to KB. /*WHY?*/
else
    Print ("Model exists") and **STOP.** /* WHY??*/
}

Good luck with midterm,
And have a good fall break!