CSE 513T Theory of Artificial Intelligence & Machine Learning February 26, 2015

Lecture 14

Lecturer: Brendan Juba *Scribe: Matt Robinson*

1 Overview

1. Declarative Knowledge and Reasoning
2. Proof Systems
3. Chaining

2 Declarative Knowledge and Reasoning

2.1 Classical Knowledge

Definition: Instance space x "the set of all possible worlds" think of as {0,1} set of all possibilities D $\subseteq$ X The attributes (or "propositions") have some interpretation, e.g. x_1 might indicate "there is a dog (in the scene) x_2 might be "there is a furry animal" For convenience, we'll write the attributes like dog(x) for x_1 and furry-animal (x) for x_2. Shorthand for state of scene indicated.

χ contains all possible settings, including those where $dog = 1$, $furry\text{-}animal = 0$. But the set of actual possibilities might only contain scenes where whenever $dog = 1$, $furry\text{-}animal = 1$ also That is the Set contains D. We say "c is valid for D" denoted $D \models c$, where c is $\{\chi \in X : \chi_1 \rightarrow \chi_2\}$.

We associate c with the representation $\chi_1 \rightarrow \chi_2$ (c(x)). These representations are called "declarative knowledge".

2.2 Knowledge Base

We've seen how to acquire knowledge from example scenes, we can also acquire declarative knowledge directly.

We refer to the collection of these representations of declarative knowledge possessed by a system as it's knowledge base (or "KB") So if KB contains $\psi_1, \psi_2, ..., \psi_k$ this corresponds to the representation $\psi_1 \wedge \psi_2 \wedge, ..., \wedge \psi_k$

Suppose we know: $dog(x) \rightarrow furry\text{-}animal(x)$ and $dog(x)$. The KB corresponding to these is

$$[dog(x) \rightarrow furry\text{-}animal(x)] \wedge dog(x) \wedge ...$$

This asserts that

1. There is a dog in the scene
2. Wherever there is a dog, there is also a furry animal

Following our notation from earlier, we will write

$$\{dog, dog \to furry\text{-}animal\} \models furry\text{-}animal \text{ or } KB \models furry\text{-}animal$$

A small KB can entail many more facts/ representations /etc. than are explicitly included. It's usually infeasible to list out all of the facts entailed by the KB explicitly.

The process of deciding whether or not a given "query" representation is entailed by a KB is called reasoning or "deductive inference"

Definition: For classes of representations $\mathscr{F}$ and $\mathscr{H}$ a reasoning task has the from given KB $\subseteq \mathscr{H}$ and a query $\varphi \in \mathscr{F}$, output "accept" or "reject"

1. An algorithm that only accepts when KB $\models \varphi$ is said to be sound
2. If an algorithm accepts φ whenever KB $\models \varphi$ we say that it is complete

Note: For a KB $\psi_1, \psi_2, ..., \psi_k$ the KB does not entail φ if and only if there is some counterexample $\chi^* \in X$ such that $\chi^* \notin \{x : \varphi(x) = 1\}$ and $\chi^* \in \{\chi(\Psi_1 \cap ... \cap \Psi_k(x) = 1\}$

i.e such that $\not\varphi(\chi^*) = 1$ and $(\Psi_1 \wedge ... \wedge \Psi_k)(\chi^*) = 1$ so deciding whether KB $\models \varphi$ is equivalent to deciding whether or not $(\neg\, \varphi \wedge \Psi_1 \wedge ... \wedge \Psi_k)$ (χ) has a satisfying argument

If we say $\mathscr{H}$ contains all OR's of ≤ 3 literals (and $\mathscr{F}$ is almost anything!) or else if $\mathscr{F}$ contains all 3 DNF representations then 3SAT (3 CNF satisfiability) is a special case of deciding entailment

Most interesting applications would be ruled out if $\mathscr{H}$ couldn't contain $\forall$OR's of ≤ 3 literals

Therefore, sound and complete reasoning for many natural classes of representations is NP-Complete

3 Proof Systems

3.1 Definition

A proof system specifies a limited family of conclusions based on a collection of syntactic rules.

Proof Systems vary in their expressive power analogously to classes of representations (that we'd denoted by ζ in PAC learning) and the use of a more expressive proof system comes at a computational cost.

We'll be comparing the expressive power and computational costs of proof systems

Definition: A proof system is given by a class of representations $\mathscr{F}$ and a relation

$$\mathrm{R} \subseteq \mathscr{F} \times \mathscr{F}^* \text{ (} \leftarrow \text{ sequences of 0 or more)}$$

A proof of $\varphi \in \mathscr{F}$ from premises $\psi_1, ...\psi_\ell \in \mathscr{F}$ is given by a sequence $\pi - \pi, \pi_2, ..., \pi\ell$ where each $\pi_i \in \mathscr{F}$ and for some subsequence $\pi_1, ..., \pi_r$ of $\pi_1, ..., \pi_i$, we have $(\pi_i, \pi'_1 ... \pi'_r) \in \mathrm{R}$ or $\pi_1 \in \Psi_1, ..., \Psi_k$ and

$$\pi_i = \varphi$$

When $\pi_i \in \mathrm{R}$, we say π_i is an axiom. Otherwise, we say it follows from a rule of inference (if it isn't one of $\Psi_1, ..., \Psi_k$)

Our proof systems will generally be sound meaning that if $(\pi_i, \pi'_i ... \pi'_r) \in \mathrm{R}$ then $\{\pi'_i ... \pi'_r\} \models \pi_i$

In this case, notice, the axioms must be tautologies (i.e, $\{\ x : \pi(x) = 1\ \} = X$)

Because entailment is transitive, a proof of φ from $\{\Psi_1, ..., \Psi_k\}$ ensures that $\{\Psi_1, ..., \Psi_k\} \models \varphi$

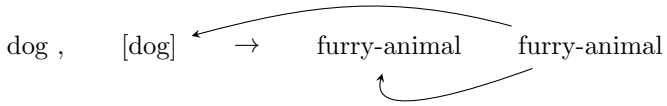
4 Chaining

A simple proof system where $\mathscr{F}$ will consist of a representation of the form $[\ell_1 \wedge ... \wedge \ell_r] \to \ell^*$ We call single literals "facts" and the $[\ell_1 \wedge ... \wedge \ell_r] \to \ell^*$ "rules" where ℓ^* is the head and the$[\ell_1 \wedge ... \wedge \ell_r]$ is the body of the rule.

We don't have axioms, and there is a single rule of inference, the (forward) chaining rule from $[\ell_1 \wedge ... \wedge \ell_r] \to \ell^*$ and $\ell_1, \ell_2 ... \ell_r$ infer ℓ^*

4.1 Example 1

If our KB contains $[dog] \to furry\text{-}animal, dog$ A chaining proof of furry animal from KB is



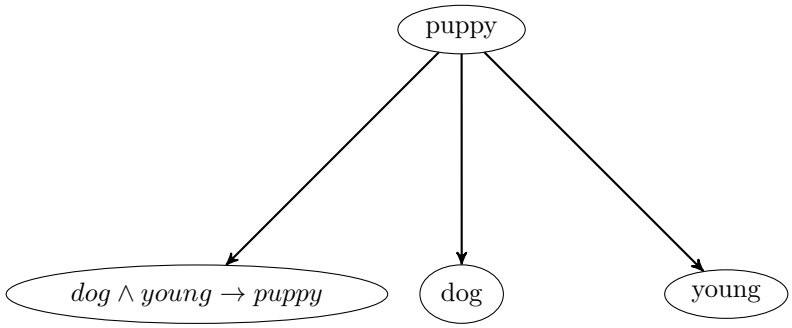
4.2 Example 2

If our KB contains $[dog \wedge young] \to puppy, dog, young$ A chaining proof of puppy from KB is:

$$dog, young, [dog \wedge young] \to puppy, puppy$$

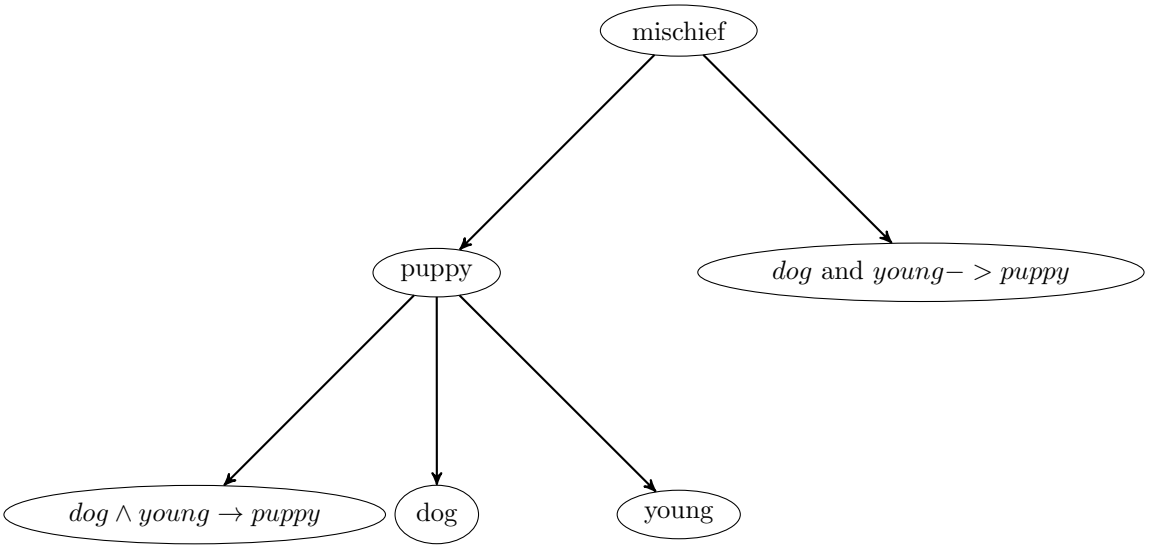
4.3

Given a proof π, there is a natural graph associated with it: the lines of the proof are the nodes, There is an edge from π_i to $\pi_j (j < 1)$ that is among the previous lines of the proof $(\pi'_1, ..., \pi'_r)$ used in a rule of inference.



4.4 Example 3

If I add $[puppy] \to mischief$, I can prove mischief. The proof's graph is



4.5

This graph is acyclic and direct
- it has a single source that is the conclusion
- the sinks are all axioms or premises

Any ordering of all the nodes π that respects the ordering of the edges is also a proof of the conclusion.

An important syntactic restriction of a proof system is when this graph is a tree, corresponding to the case where intermediate derivations aren't reused we say that the proof is "tree-like".

5 Next Time: Resolution

Resolution is a richer proof system in which $\mathscr{F}$ is the class of disjunctions (that we'll call clauses)

- Has no axioms
- It has a single rule of inference called cut given two clauses of the form

$$x_1 V \text{ and } \neg x_1 V$$

Cut infers $c_1 \vee c_2$ (removing the "complementary pair" x_1 and x_2) taking the or of the rest

Question to think about: Why is this sound?