Inference in first-order logic

• Our goal is to prove that $\text{KB}$ entails a fact, $\alpha$
  – We use logical inference
    – Forward chaining
    – Backward chaining
    – Resolution

• All three logical inference systems rely on search to find a sequence of actions that derive the empty clause

Search and forward chaining

• Start with $\text{KB}$ full of first-order definite clauses
  – Disjunction of literals with exactly one positive
    • Equivalent to implication with conjunction of positive literals on left (antecedent / body / premise) and one positive literal on right (consequent / head / conclusion)
  • Propositional logic used Horn clauses, which permit zero or one to be positive
    – Look for rules with premises that are satisfied (use substitution to make matches) and add conclusions to $\text{KB}$

Search and forward chaining

• Would other search methods work?
  – Yes, this technique falls in standard domain of all searches
Search and backward chaining

• Start with KB full of implications
  – Find all implications with conclusion matching the query
  – Add to fringe list the unknown premises
    • Adding could be to front or rear of fringe (depth or breadth)

Search and backward chaining

Depth First
  • Are all the premises of I satisfied? No
  – For each (C E G H) are each of their premises satisfied?
    • C? no, put its premises on fringe
    – For each (A and E) are their premises satisfied?
      A… yes
      E… no, add premises for each B and D
      B… yes
      D… yes
      – E, G, H… yes

Breadth First
  • Are all the premises of I satisfied? No
  – For each (C E G H) are each of their premises satisfied?
    • C? no, put its premises on fringe end
    – E? no, put its premises on fringe end
    – G, H… yes
    – Are C’s premises (A E) satisfied?
      A… yes
      E… no, add premises
    – Are E’s premises (B D) satisfied?
      Yes
    – Return to C and I

Search and backward chaining

Backward/forward chaining

• Don’t explicitly tie search method to chaining direction

Inference with resolution

– We put each first-order sentence into conjunctive normal form
  • We remove quantifiers
  • We make each sentence a disjunction of literals (each literal is universally quantified)
  – We show KB ^ α is unsatisfiable by deriving the empty clause
    • Resolution inference rule is our method
      – Keep resolving until the empty clause is reached

Resolution

• Look for matching sentences
  – Shared literal with opposite sign
    • Substitution may be required
  – [Animal (F(x)) V Loves (G(x), x)] and [-Loves (u,v) V ~Kills (u, v)]
    • F(x) = animal unloved by x
    • G(x) = someone who loves x
Resolution

- What does this mean in English?
  - $[\text{Animal (F(x)) V Loves (G(x), x)}]
  - $F(x) = \text{animal unloved by x}$
  - $G(x) = \text{someone who loves x}$
  - $[\neg\text{Loves (u,v) V \neg\text{Kills (u, v)}]$

- For all people, either a person doesn't love an animal or someone loves the person
- Nobody loves anybody or nobody kills anybody

Example

$\neg\text{American}(x) \vee \neg\text{Weapon}(y) \vee \neg\text{Sale}(x, y, z) \vee \neg\text{Hostile}(z) \vee \text{Criminal}(x),$
$\neg\text{ Missile}(x) \vee \neg\text{Nano}(x, y) \vee \text{Sale}(\text{West}, x, \text{Nano}),$
$\text{Enemy}(x, \text{America}) \vee \text{Hostile}(x),$
$\neg\text{ Missile}(x) \vee \neg\text{Weapon}(x),$
$\text{Nano}(\text{Nano}, M_1), \text{ Missile}(M_1),$
$\text{American(\text{West}), Enemy(\text{Nano, America}).}$

Inference with resolution

- What resolves with what for proof?
  - Unit preference
    - Start with single-literal sentences and resolve them with more complicated sentences
    - Every resolution reduces the size of the sentence by one
    - Consistent with our goal to find a sentence of size 0
  - Resembles forward chaining

Resolution example

- $[\text{Animal (F(x)) V Loves (G(x), x)}]$ and $[\neg\text{Loves (u,v) V \neg\text{Kills (u, v)}]$
  - Loves and $\neg$Loves cancel with substitution
    - $u=G(x)$ and $v=x$
  - Resolvent clause
    - $[\text{Animal (F(x)) V \neg\text{Kills (G(x), x)}]}$

- Set of support
  - Build a special set of sentences
  - Every resolution includes one sentence from set
  - New resolvent is added to set
  - Resembles backward chaining if set of support initialized with negated query
Theorem provers

- Logical inference is a powerful way to “reason” automatically
  - Prover should be independent of KB syntax
  - Prover should use control strategy that is fast
  - Prover can support a human by
    - Checking a proof by filling in voids
    - Person can kill off search even if semi-decidable

Practical theorem provers

- Boyer-Moore
  - First rigorous proof of Gödel Incompleteness Theorem
- OTTER
  - Solved several open questions in combinatorial logic
- EQP
  - Solved Robbins algebra, a proof of axioms required for Boolean algebra
    - Problem posed in 1933 and solved in 1997 after eight days of computation

Practical theorem provers

- Verification and synthesis of hard/soft ware
  - Software
    - Verify a program’s output is correct for all inputs
    - There exists a program, P, that satisfies a specification
  - Hardware
    - Verify that interactions between signals and circuits is robust
      - Will CPU work in all conditions?
    - There exists a circuit, C, that satisfies a specification

Statistical Learning Methods

- Chapter 20
  - Statistical learning (Bayes, maximum likelihood)
  - Hidden variables (expectation maximization, Markov models)
  - Instance-based (Nearest neighbor)
  - Neural networks

Rational agents

- Up until now
  - Many rules were available and rationality was piecing rules together to accomplish a goal
    - Inference and deduction
- Now
  - Lots of data available (cause/effect pairs) and rationality is improving performance with data
    - Model building, generalization, prediction

How early will my son be born?

- Logic from first principles
  - I think he will be born tomorrow
    - 20 literals corresponding to 20 dates
    - Well-fed (mom(x)) => late(x)
    - late(x) ^ impatient(father(x)) => thisWeekend (x)
    - late(x) ^ impatient(mother(x)) => tomorrow(x)
    - ...
How early will my son be born?

- Statistical Learning
  - Histogram of births
  - Data from family tree
  - Multidimensional correlations between early and ethnicity
  - ...

Function Approximator

- Build a function that maps input to output
  - Start with a model of function
  - Use statistics to set values of coefficients
    - Pick $m$ and $b$ such that line defined by terms minimizes the sum of distances between each observed $(x, y)$ and $(x, f(x))$

\[ f(x) = mx + b = y \]

Slightly more complicated

- Parabola
  - Select $a$, $b$, $c$
  - Goal is $y - ax^2 - bx - c = 0$
    - If we have three points and three unknowns we can solve
    - If we have more points we must use another technique

Mappings

- These function approximators are mappings
  - They map inputs to outputs
    - We hope the outputs match similar observations
  - The mappings become better with more information
  - This is what neural networks do
    - But the beauty of neural networks is in how they do what they do

Neural Networks

- Biologically inspired
  - We have neurons in our bodies that transmit signals based on inputs
    - Internal dynamics dependent on chemical gradients
    - Connections between neurons are important
  - Tolerates noisy input
  - Tolerates partial destruction
  - Perform distributed computation

Neural Networks

- Synthetic
  - A neural network unit accepts a vector as input and generates a scalar output dependent on activation function
  - Links within network controlled through weights