Search

Finding Solutions in Life since Brutus

EPGY 2011
Computer Science: Artificial Intelligence
Lekan Wang

Mathematical Proof
Search
Uninformed Search
Informed Search

Proof

- Series of steps that logically follow from step to conclusion
- Showing validity
- To disprove, just show counterexample

Enumeration

- "Brute Force"
- Just list every possibility
- Obviously only applies to finite, enumerable sets
- Useful for certain applications, like logic (truth tables)

Direct

- Start with premises, and get to goal
- For equivalence, remember to prove both ways

Proof by Contradiction

- Prove that the opposite of our statement is false
- Assume goal is false
- Prove a contradiction
- We essentially used this when doing automated logical resolution
- Examples
  - Infinitude of Primes
  - Sqrt(2) is irrational

Induction

- "Recursion in math"
- Requires partial ordering (ordered, and bounded at one end)
- Example
  - $0+1+\ldots+n = n(n+1)/2$

Mathematical Proof

Search

Uninformed Search

Informed Search

Search

- High-level definition: Examining future options to determine what immediate action should be taken
- Assumptions
  - Environment Observable (not hidden)
  - Environment known
  - Deterministic actions

Problem Formulations

- States
- Initial State
- Actions
- Transition Model
- Goal Test
- Path Cost
  - Step Cost

Search

- Better definition: Looking for a *sequence of actions* to take that achieves the *goal*. That sequence of actions is called the *solution*.
- Initial State, actions, transition model are, together, called the *state space*.
  - All reachable states from the initial state
- All state spaces can be represented as a graph

Search Terminology

- Incremental (start with nothing, and add states)
- Complete-state (start with everything) and rearrange, or delete

Search Tree

- All searches can be represented as a search tree of nodes
- Nodes are not states. Nodes represent a state in the search
- Expanding nodes
- Leaf nodes of search together, is the *frontier*
- Can keep *explored set* to not visit repeat.

General Graph Search

- See Russell/Norvig figure 3.7
- Different searches all about how you choose next leaf to expand.
- Nodes need to store
  - Corresponding state
  - Parent
  - Action taken by parent to get here
  - Path Cost (g(n))

Breath-First Search (BFS)

- Choose next node by putting nodes in Queue
- Ignores all costs

Depth-First Search (DFS)

- Replace the queue in BFS with a stack

Uniform-Cost Search

- Also called Dijkstra's Algorithm
- Expand node with lowest g(n)

Iterative Deepening DFS

- Go one layer at a time, doing DFS every time
- Repeats nodes

Bidirectional Search

- Start search from both start and goal.
- This works because if *e* is avg number of edges at a node,
  - e^t nodes expanded in *t* steps.
  - If 2<e, then $2(e^t) < e(e^t)$

Mathematical Proof

Search

Uninformed Search

Informed Search

Mathematical Proof

Search

Uninformed Search

Informed Search

Informed Search

- Uses a *heuristic function, h(n)*
  - Intuition about the world
  - Estimated cost of cheapest path from the state at node *n* to goal
  - Let *C=g+h*
  - *h**=oracle heuristic
  - *C** = Optimlal cost

Detour: Distance

- *d* is a distance iff *d*:X,Y->**R**, s.t.
  - *d(p,q) >= 0*
  - *d(p,q) = 0 iff p=q*
  - *d(p,q) = d(q,p)* Commutativity
  - *d(p,q) <= d(p,r)+d(r,q)* Triangle Inequality

Detour: Distance

- Euclidean Distances
  - Square roots of sum of squares (L_2 norm)
  - Manhattan distance (L_0 norm)
  - L_{inf} norm (Chebyshev distance) = $max_i(abs(p_i – q_i))$
- Non-Euclidean
  - Hamming distance
  - Edit distance
  - Cosine distance
  - Jaccard distance

Evaluation

- Completeness: Will check all nodes to find solution
- Optimality: The solution returned will be optimal
- Optimally Efficient: For another algorithm using same cost function and same heuristic, you cannot expand fewer nodes.

Best-First Search

- Give every node an *f(n)* to calculate a priority.
- Pick the one with the best priority
- With uniform cost search, notice *f(n)=g(n)*
- Greedy Best-first search
  - Intuition: Expand node closest to goal
  - Complete: Yes
  - Optimal: No

A* Search

- Set *f(n) = g(n) + h(n)*
- Intuitively, f(n) is estimated cost from initial state to goal through the current node
- Conditions
  - *h(n)* must be *admissible heuristic*
  - Consistency (monotonicity)
  - *h(n) <= c(n,a,n')+h(n')*

A*

- Optimal
- Complete
- Optimally Efficient
- However, can be costly

Alternatives

- IDA* (Iterative Deepening A*)
- RFBS (Recursive BFS)
- SMA (Simple Memory-Bounded A*)

Search

- Domination
  - For any node *n*, if $h_2 >= h_1$ then *h2* dominates h_1.
  - Since we can't pass *C**, dominant heuristics are good
  - A weak heuristic takes us closer to uniform-cost search
- Generating heuristics
  - Multiple heuristics: can take $max\{h_1, h_2, h_3, \ldots\}$
  - Cost of optimal solution to a relaxed problem is an admissible heuristic fo the original problem