1 handout: slides

EOLQs

Search

Contents

This particular pattern of molecules known as a 'human being' has evolved an amazing depth of consciousness: an ability to internally model the reality beyond the senses, to imagine futures that have never happened, to use language, to use rationality to build and test theories about our universe, to become self-aware.
—Jeff Lieberman (artist, roboticist)

Cognitive Science

The ability to think is perhpas the most distinctive of human capacities. Typically, thinking involves mentally representing some aspects of the world (including aspects of ourselves) and manipulating these representations or beliefs so as to yield new beliefs, where the latter may aid in accomplishing a goal.
—Edward E. Smith (Psychology, U Michigan)

The ability to solve problems is one of the most important manifestations of human thinking. ... We might therefore suspect that problem solving depends on general cognitive abilities that can potentially be applied to an essentially unlimited range of domains.
—Keith Holyoak (Psychology, UCLA)

Formalizing Problem Solving

State: hypothetical world state
Operators: actions that modify world
Goal: desired state or test

(Herbert Simon and Allen Newell, "Computer simulation of human thinking and problem solving", 1961)

Representation

VW search space

VW state space

MC representation

Basic Algorithms

________-First Search

open $\leftarrow$ an ordered list containing just the initial state.
Loop
    If open is empty,
        then return failure.
    Node $\leftarrow$ Pop(open).
    If *Node* is a goal,
        then return *Node* (or path to it).
        else
            Children $\leftarrow$ **Expand** (*Node*).
            Add *Children* to front of open.

Evaluating DFS

Assume branching factor b and solution at depth d.

Completeness:
Time:
Space:
Admissibility:

Breadth-First Search

open $\leftarrow$ an ordered list containing just the initial state.
Loop
  If open is empty,
    then return failure.
  Node $\leftarrow$ Pop(open).
  If *Node* is a goal,
    then return *Node* (or path to it).
    else
      Children $\leftarrow$ **Expand** (*Node*).
      Add *Children* to end of open. $\leftarrow$

Evaluating BrFS

Assume branching factor b and solution at depth d.

Completeness:
Time:
Space:
Admissibility:

Uniform-Cost Search

open $\leftarrow$ an ordered list containing just the initial state.
Loop
  If open is empty,
    then return failure.
  Node $\leftarrow$ Pop(open).
  If *Node* is a goal,
    then return *Node* (or path to it).
    else
      Children $\leftarrow$ **Expand** (*Node*).
      Merge *Children* into open, keeping sorted by path cost. $\leftarrow$

Dealing with Graphs

1. Check for cycles with ancestors
2. Maintain `closed` list (hash table) to detect duplicates

Comparison

Algorithm	Time	Space	Complete	Admissible
Depth-first	b^m	bm	If $m \geq d$	No
Breadth-first	b^d	b^d	Yes	If ops cost 1
Uniform-cost	b^d	b^d	Yes	Yes

branching factor	b
maximum depth	m
solution depth	d

Time and Space for BrFS/UCS

Assume $b = 10$; 100,000 nodes/sec; 100 bytes/node.

Sol. depth	Nodes	Time	Space
1	11	.11 msec	1.1 Kb
2	111	1.1 msec	11 Kb
4	11,111	.11 sec	1 Mb
6	10^6	11 sec	111 Mb
8	10^8	18 min	11 Gb
10	10^{10}	31 hours	1 Tb
12	10^{12}	128 days	111 Tb
14	10^{14}	35 yrs	11 Pb

Search Conundrum

Breadth-first uses b^d space

but complete and admissible

Depth-first complete only if *limit* $> d$, not admissible

but bd *space*

How can we get the best of both?

Break

- textbook
- piazza
- asst 1
- sources
- recitation
- office hours

A Clever Algorithm

Iterative Deepening Search

for $d = 1$ to ∞ do
    depth-first search to level d
    if it succeeds
        then return solution

Could this possibly be efficient?

Evaluating IDS

Assume branching factor b and solution at depth d.

Completeness:
Time:
Space:
Admissibility:

Nodes Generated by IDS

$b = 2$

d	at d	in prev.	total	IDS	% of opt.
0	1	0	1	1	100.0
1	2	1	3	4	133.3
2	4	3	7	11	157.1
3 4	16	15	31	57	183.9

$b = 10$

d	at d	in prev.	total	IDS	% of opt.
0	1	0	1	1	100.0
1	10	1	11	12	109.1
2	100	11	111	123	110.8
3 4	10000	1111	11,111	12,345	111.1

Nodes Generated by IDS

$$b^d + 2b^{d-1} + 3b^{d-2} + ... + (d-1)b^2 + db$$

$$\approx b^d(\frac{b}{b-1})^2$$

EOLQs

EOLQs

Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

Thanks!