CS 188, Fall 2015

Intelligent Agents

Rationality: achieving maximum utility by some pre-defined metric or set of goals/intentions.
depends on the usefulness of the choice and not on the process that led to that choice
e.g., rational process for playing tic-tac-toe by tabulating game states
has no decision process, but acts rationally
An agent has sensors which take in percepts and actuators which effect actions
a percept is a set of perceptual inputs at a fixed point in time
a percept sequence is composed of percepts
An agent function is $f : P^* \to A$ where P^* is the percept sequence and A the set of actions
the agent program is the specific architecture which implements this function
not all agent functions can be implemented by some agent program
e.g. halting problems, NP-hard problems, "too-large" problems (e.g. chess)
Performance measure: an objective criterion for the success of behavior of an agent
Rational agent: maximizes expected performance measure by its actions
given the prior knowledge and percept sequence available to it
e.g. vacuum world: 2 squares, could be dirty or clean
action function: suck if dirty, move if clean
under the measure of most clean squares/time period, this is rational
if we seek to minimize movements as well, this is irrational
Autonomy: the ability to function beyond the prior knowledge of the designer

The task environment: performance measure, environment, actuators, and sensors
partially vs. fully observable (perceive all aspects relevant to choice of action)
stochastic vs. deterministic (next state a function only of current state, action)
strategic: deterministic but for the actions of other agents
episodic vs. sequential (current decision could affect future decisions)
static vs. dynamic (environment can change while agent deliberates)
semidynamic: environment doesn't change with time but performance score does
semidynamic e.g. chess with a clock
discrete vs. continuous (can apply to state, time, percepts/actions)
single vs. multiagent: e.g. competitive multiagent, cooperative multiagent

Agent structure
the agent program only takes in the current percept
the agent function maps from the entire percept history
Simple reflex agents
This is where we at.

Agent types (increasing generality/complexity):
simple reflex

state-based reflex agents
goal-based agents
utility-based agents
Two of these are reflexive, the next two are planning-based.

9/3

Planning agents predict consequences of actions $\rightarrow$ transition model.
Deliberativeness: can generate a complex plan or a simple one and correct rapidly.
A search problem consists of a state space, a set of allowable actions, a transition model (corresponding to results), a step cost function, a t_0 and a goal test.
Solution: A_n that transforms a t_0 into a goal state.
A real world state is highly general.
A search state is specific to the problem, focusing on the vital details of the environment.
Can describe the state space using a directed graph. Note that each state appears only once in the graph. Edges can have costs associated with them.
Search trees incorporate temporal direction.
Implement nodes with state, parent, action, cost of action (path-cost)
Depth-first-search, breadth-first-search, iterative deepening (DFS with limit 1, DFS with limit 2, etc.)
Uniform-cost-search: expand frontier at cheapest node first. A* can prove is optimal. However, uses no information about goal.
Tree searches can often lead to excess/repeated work. Can correct by making a check whether or not potential actions lead to an explored state.
Graph search exists in a finite space, memory $\propto$ runtime.

9/8

Heuristic estimates how close a state is to the goal
Greedy search expands nodes based upon a given heuristic
Uniform-cost search orders by path cost, i.e. backwards cost, g(n)
Greedy search orders by proximity to goal, i.e. forward cost, h(n)
An A^* algorithm orders by $f(n) = g(n) + h(n)$
Key point: an A^* search must have an effective heuristic to work
An admissible heuristic satisfies $0 \leq h(n) \leq h^*(n)$, with $h^*(n)$ the true cost
A^* search with an admissible heuristic is optimal
A heuristic h_a is dominant over h_c, $h_a \geq h_c$ if $\forall n \; h_a(n) \geq h_c(n)$
A dominant heuristic of an admissible heuristic, if still admissible, is preferable
A maximum over admissible heuristics is a dominant heuristic, and still admissible

9/10

9/15