Review of AI

Review

Adapted from slides kindly shared by Stuart Russell

Appreciations

◇ Straightforward communication

Share some of yours?

Announcements

Project P4: Ghostbusters out, due Dec 19 - but understand it before the final exam....

Still time to participate on Piazza and improve your score there. But no "me too!" posts....

It is still an option to re-submit previous projects for 1/2 credit on incremental improvement over original

NB: Final Exam in ECCR 200 Tuesday 2012-12-18 04:30 PM - 07:00 PM

Final: closed book; 2 pages of notes OK; non-programmable calculator optional; 50% on new material

See Piazza for topics, practice exams

Outline

- What to study
- Big picture
- Review major topics

What to study

- Study material we spent a lot of time on
- Try practice exams
- Review homeworks

Big picture

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances

The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

Environments are categorized along several dimensions:
observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist:
reflex, reflex with state, goal-based, utility-based

Problem types

Deterministic, fully observable $\Longrightarrow$ single-state problem

Agent knows exactly which state it will be in; solution is a sequence

Non-observable $\Longrightarrow$ conformant problem

Agent may have no idea where it is
Solution (if any) is a sequence in belief space

Nondeterministic and/or partially observable $\Longrightarrow$ contingency problem

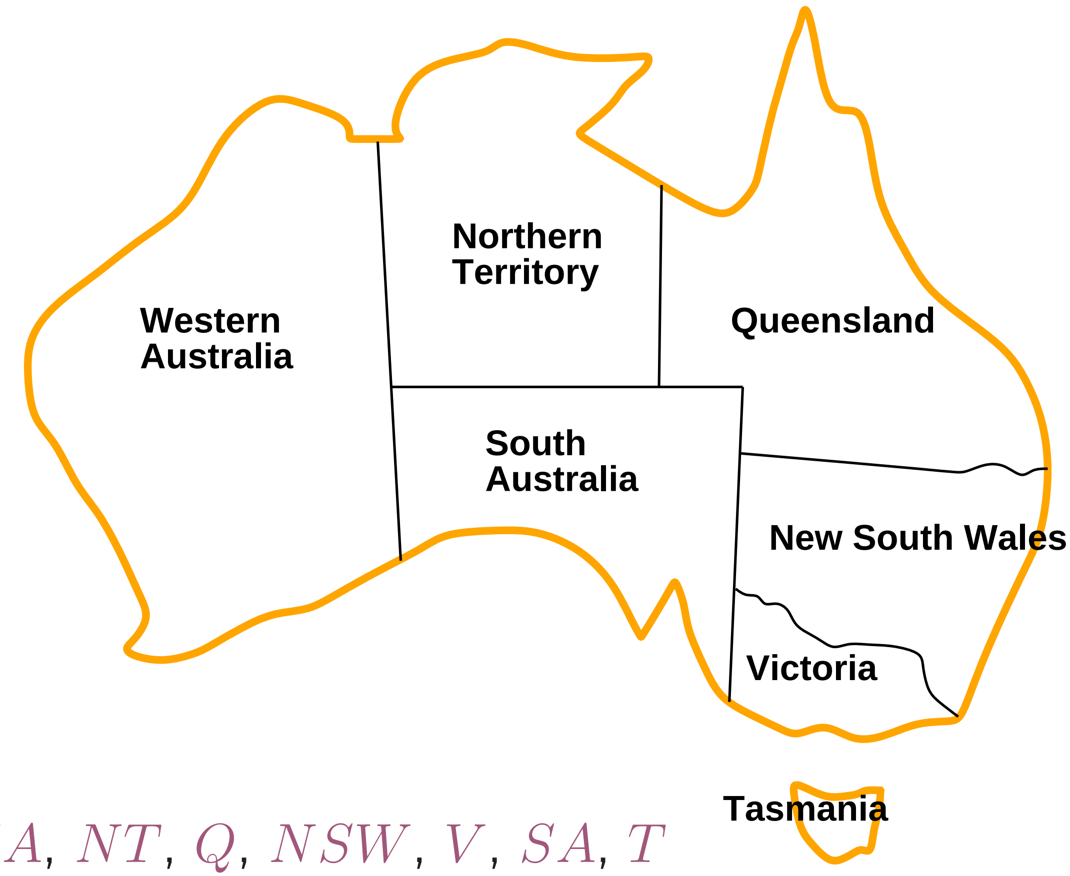
percepts provide **new** information about current state
solution is a contingent plan or a policy
often **interleave** search, execution

Unknown state space $\Longrightarrow$ exploration problem ("online")

Search

- BFS, DFS, UCS, A*, Greedy search (tree and graph)
- Search algorithms' strengths and weaknesses
- Properties: completeness, optimality
- Admissibility and consistency for A*
- Be able to formulate search problems and create heuristics

CSP example: Map-Coloring



Variables WA, NT, Q, NSW, V, SA, T

Domains $D_i = \{red, green, blue\}$

Constraints: adjacent regions must have different colors

e.g., $WA \neq NT$ (if the language allows this), or

$(WA, NT) \in \{(red, green), (red, blue), (green, red), (green, blue), \ldots\}$

Constraint Satisfaction Problems

- Basic definitions and solution with DFS (Backtracking search)
- Efficiency: ordering and checking
  - Variable choice: Minimum Remaining Values
  - Value choice: Least constraining value
  - Forward checking
  - Constraint propagation, e.g. arc consistency
- Conditions under which CSPs are efficiently solvable: tree structure etc.
- Local search for CSPs: min-conflicts
- Be able to formulate CSPs

Games

- Representation, game trees
- Minimax search
- Alpha-beta pruning
- Expectimax search
- Evaluation functions
- Metareasoning

MDP trees vs Expectimax

Markov Decision Processes - a family of non-deterministic search problems

States $s \in S$, start state s_0, actions $a \in A$

Transition function, like Successor function

<u>Model</u> $T(s, a, s') \equiv P(s'|s, a) =$ probability that a in s leads to s'

Q-states, like choice nodes

<u>Reward function</u> $R(s)$ (or $R(s, a)$, $R(s, a, s')$), vs prize at the end

MDPs - Nondeterministic search

- The maximum expected utilitiy (MEU) principle
- Reflex agents and policies
- Markov decision process definition
- Reward functions, values and q-values
- Bellman Equations:
- $V^*(s) = max_a Q^*(s,a)$
- $Q^*(s,a) = \Sigma_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$
- Value and policy iteration
- Be able to formulate a problem as an MDP

Reinforcement Learning

- Exploration vs exploitation
- Epsilon-greedy
- Model-based and model-free learning
- Temporal-Difference value learning / Q-learning
- Weighted features and linear value function approximation

Probability

- Conditional probabilities, product rule, chain rule, Bayes rule
- Joint, conditional and marginal distributions
- Independence and conditional independence
- Inference by enumeration from joint distributions

Bayes' Nets

- Representation and semantics
- Building joint distributions from conditional probability tables
- Inference from joint distributions
- Bayes net $\Rightarrow$ Joint $\Rightarrow$ Query
- Conditional independence and d-separation
- Variable elimination
- Sampling / approximate inference
- Formulating Bayes' nets for problems

VPI

- Drawing and reasoning about decision networks
- Finding actions that maximize expected utilities
- Manipulating Bayes' nets to compute conditional probabilities
- Computing VPI of a random variable

Hidden Markov Models

- HMM structure and Bayes' net properties
- Forward algorithm, computing belief distributions