CS440/ECE448 Lecture 31: Exam 2 Review

Mark Hasegawa-Johnson, 4/2024

Outline

- How to take the exam
- Topics
  - Search, MDP, Minimax, Expectiminimax, Game Theory
  - Unification, Vector Semantics, Robotics
- Sample problems

How to take the exam

- Reserve a time at https://us.prairietest.com/
- Show up at the appointed time to take the exam
- There will be 8 multiple choice questions:
  - search (lectures 18-19), Markov decision process (lecture 20), minimax (lecture 21), expectiminimax (lecture 22), static game theory (lecture 23)
  - logic (lectures 26-27), vector semantics (lectures 28-29), robotics (lec 30).
- Not covered: repeated games (lec24), privacy (lec25)

Search

- BFS, DFS, UCS, A*
- Admissible heuristic: $\hat{h}(n) \leq h(n)$
- Consistent heuristic: $\hat{h}(n) - \hat{h}(m) \leq h(n, m)$
- $\hat{h}(n) = 0$ is a valid heuristic (equal to UCS), but usually we want to invent an $\hat{h}(n)$ as large as we can, subject to one of the two constraints above (depending on whether we want to re-open closed nodes).

Markov Decision Processes

- Bellman equation:

$$u(s) = r(s) + \gamma \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s')$$

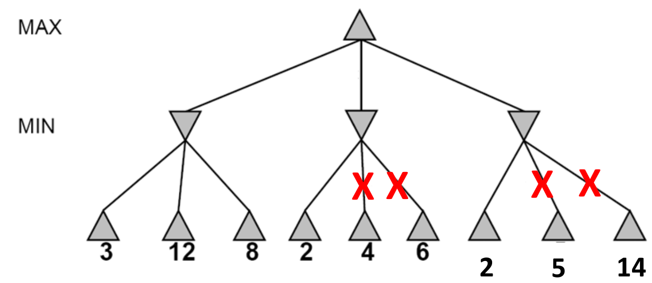
- Value iteration:

$$u_i(s) = r(s) + \gamma \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u_{i-1}(s')$$

- Policy iteration:

$$u_i(s) = r(s) + \gamma \sum_{s'} P(S_{t+1} = s' | S_t = s, \pi_i(s)) u_i(s')$$

$$\pi_{i+1}(s) = \operatorname*{argmax}_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u_i(s')$$



Minimax

- Alternating two-player zero-sum games
  - $\wedge$ = a max node, $\vee$ = a min node
- Minimax search
  - $v(s) = \max_a v(\text{child}(s, a))$ or $v(s) = \min_a v(\text{child}(s, a))$
- Limited-horizon computation and heuristic evaluation functions

$$v(s) = w_1 f_1(s) + w_2 f_2(s) + \cdots$$

- Alpha-beta search
  - Min node can update beta, Max node can update alpha
  - If beta ever falls below alpha, prune the rest of the children
- Computational complexity of minimax and alpha-beta
  - Minimax is $O\{b^d\}$. With optimal move ordering, alpha-beta is $O\{b^{d/2}\}$.

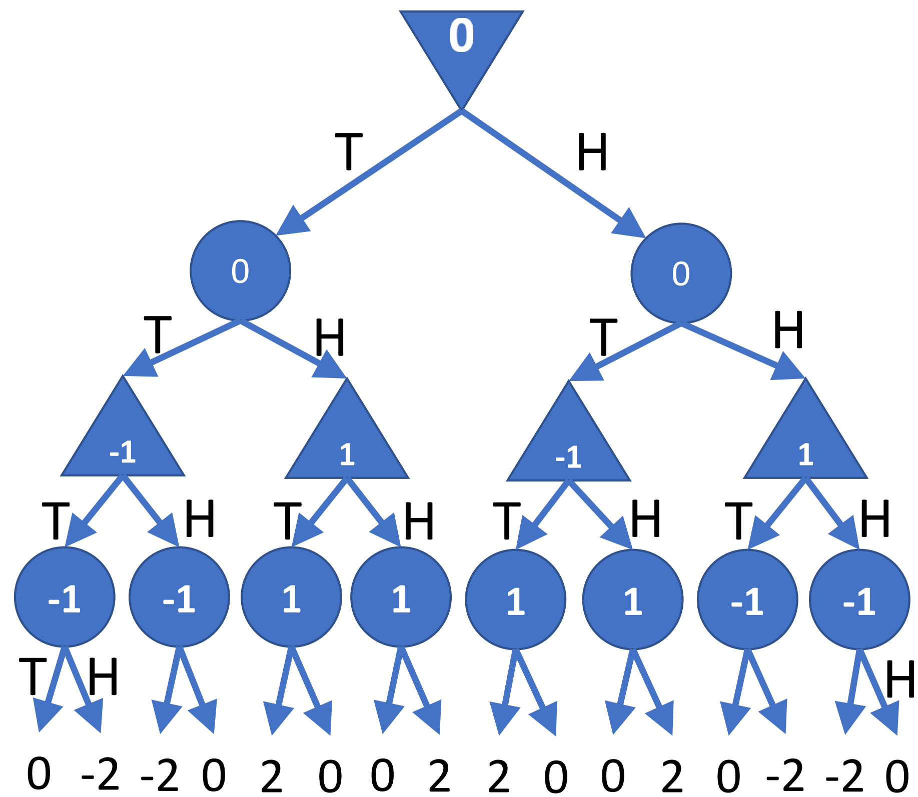
Expectiminimax

Bellman equation = expectimax:

$$u(s) = \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s')$$

Expectiminimax:

$$u(s) = \begin{cases} \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s') & s \text{ is a max state} \\ \min_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s') & s \text{ is a min state} \end{cases}$$



Game Theory

- Dominant strategy
  - a strategy that's optimal for one player, regardless of what the other player does
  - Not all games have dominant strategies
- Nash equilibrium
  - an outcome (one action by each player) such that, knowing the other player's action, each player has no reason to change their own action
  - Every game with a finite set of actions has at least one Nash equilibrium, though it might be a mixed-strategy equilibrium.
- Pareto optimal
  - an outcome such that neither player would be able to win more without simultaneously forcing the other player to lose more
  - Every game has at least one Pareto optimal outcome. Usually there are many, representing different tradeoffs between the two players.
- Mixed strategies
  - A mixed strategy is optimal only if there's no reason to prefer one action over the other, i.e., if $0 \leq p \leq 1$ and $0 \leq q \leq 1$ such that:

$$(1-p)w + px = (1-p)y + pz$$
$$(1-q)a + qc = (1-q)b + qd$$

Logic

- Logic:
  - $\neg$ (not) , $\wedge$ (and) , $\vee$ (or), $\Longrightarrow$ (implies), $\Longleftrightarrow$ (equivalent)
  - First-Order Logic: $\exists x: F(x)$ (there exists), $\forall x: F(x)$
- Proving "there exists" theorems: find an x that satisfies the statement
- Variable normalization: each rule uses a different set of variable names
- Unification: Find a substitution $S: \{\mathcal{V}_P, \mathcal{V}_Q\} \rightarrow \{\mathcal{V}_Q, C\}$ such that $S(P) = S(Q) = U$, or prove that no such substitution exists
- Forward-chaining: Search problem in which each action is a unification, and the state is the set of all known true propositions
- Backward-chaining: Search problem in which each action is a unification, and the state is the goal (the proposition whose truth needs to be proven)

Vector Semantics

- Context bag-of-words (CBOW), generative loss:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-c, j\neq 0}^{c} \ln \frac{\exp\left(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)}$$

- Skip-gram, contrastive loss:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{\boldsymbol{v}\prime\in\mathcal{D}_+(w_t)} \ln \frac{1}{1+e^{-\boldsymbol{v}\prime^T \boldsymbol{v}_t}} + \sum_{\boldsymbol{v}\prime\in\mathcal{D}_-(w_t)} \ln \frac{1}{1+e^{\boldsymbol{v}\prime^T \boldsymbol{v}_t}}\right)$$

- Attention

$$\boldsymbol{c}_i = \boldsymbol{V}^{\boldsymbol{T}} \mathrm{softmax}(\boldsymbol{K}\boldsymbol{q}_i) = \sum_t \frac{\exp\left(\boldsymbol{q}_i^T \boldsymbol{k}_t\right)}{\sum_\tau \exp\left(\boldsymbol{q}_i^T \boldsymbol{k}_\tau\right)} \boldsymbol{v}_t$$

- Self-attention, Multi-headed attention, Cross-attention, and Masked attention
- Positional encoding: $\boldsymbol{x}_t \mathrel{+}= \left[\cos\left(\frac{\pi t}{T}\right), \cdots, \sin\left(\frac{\pi D t}{2T}\right)\right]^T$

Robotics

- Workspace (e.g., $\boldsymbol{w} = [x, y]^T$) vs. Configuration space (e.g., $\boldsymbol{q} = [\theta_1, \theta_2]^T$)
- Path planning: shortest path in configuration space
  - First, map obstacles from workspace into configuration space
  - Visibility graph: states=vertices of obstacles in configuration space
  - Rapid Random Trees (RRT): states=random, resampled near the best path after every iteration
- Trajectory control
  - Time scaling: Constraints on motor torque, workspace velocity
  - Proportion-Integral-Derivative (PID) controller: Smooth out oscillations
  - Model predictive control: Plan for the possibility of error