Pacman Contest

Challenges:
- Long-term strategy
- Multiple agents
- Adversarial utilities
- Uncertainty about other agents' positions, plans, etc

- 60 submissions (over half the class!)
  - Creative names: BalsamicVinegarOfJustice, BigUtilusMaximusAgents, ...
  - Creative methods: tracking, learning, search...
- 45 qualifiers (a third of the class!)
- Amazing work by everyone!
- Final Tournament
  - Double elimination seeded by round-robin
  - Required 15 CPUs for almost a day
  - Final matches: now!

For Third Place

For First / Second Place
But… Wait!

Seed–1:–Chris–Berner Zelam Ngo


Qualifying–Agents


Seed–3:––Andrea–Goh (Skynet)


Scary–AI–from–the–future

Results

Double Elimination

1st place:–Nils–Reimer

2nd place:–Nick–/–Haotian

3rd place:–Kevin–/–Tyler

Round Robin

1st Place:–Chris–Berner /–ZelamNgo

2nd Place:–Nick–/–Haotian

3rd Place:–Andrea–Goh

Combined Results

1st place:,–Nils,Chris /–Z

2nd place:–Nick–/–Haotian

3rd place:––Kevin–/–Tyler, Andrea

…–and–congratulations–to–All!

…and–Congratulations–to–All!

Amazing work by everyone

• Record number of entries (60 teams)
• Record number of qualifications (45!)
• Lots of mutual support on newsgroup / office hours…

You should all be proud of what you’ve accomplished!

Example:–Stratagus

Stratagus (similar to Starcraft, etc):–example–of–a–large–RL–task

Stratagus is hard for reinforcement learning algorithms

• $> 10^{10}$ states
• $> 10^{30}$ actions at each point
• Time horizon $= 10^4$ steps

Stratagus is hard for human programmers

• Typically takes several person-months for game companies to write computer opponent
• Still, no match for experienced human players, very fragile
• Programming involves much trial and error

Hierarchical RL

• Humans break up the task into a multi-level sketch using a partial program
• Learning algorithm fills in the details

[From Bhaskara Marthi’s thesis, Berkeley]

Solution:–Hierarchical Learning

(defun top ()
  (loop
    (choose
      (gather-wood)
      (gather-gold))))

(defun gather-wood ()
  (with-choice
    (dest *forest-list*)
    (nav dest)
    (action ‘get-wood)
    (nav *base-loc*)
    (action ‘dropoff)))

(defun gather-gold ()
  (with-choice
    (dest *goldmine-list*)
    (nav dest)
    (action ‘get-gold)
    (nav *base-loc*)
    (action ‘dropoff)))

(defun nav (dest)
  (until (= (pos (get-state)
      dest)
    (with-choice
      (move '(N S E W NOOP)
        (action move))))
    (action ‘dropoff)))
Hierarchical RL

- Solution: hierarchical planning and learning
  - Define a hierarchical MDP
  - Each level has Q-functions (one per choice)
  - Learning happens at all levels, all at once

- State-of-the-art
  - Still not very good at the strategic elements (high level strategy)
  - Very good at balancing resources (mid-level allocation)
  - Excellent at lowest levels of control

Pacman: Beyond Simulation?

Students at Colorado University: http://pacman.elstonj.com

Bugman?

- AI = Animal Intelligence?
  - Wim van Eck at Leiden University
  - Pacman controlled by a human
  - Ghosts controlled by crickets
  - Vibrations drive crickets toward or away from Pacman’s location

Where to go next?

- Congratulations, you’ve seen the basics of modern AI
  - … and done some amazing work putting it to use!

- How to continue:
  - Robotics / vision / IR / language: cs189
  - Machine learning: cs281a / cs281b
  - Cognitive modeling: cog sci 131
  - Vision: cs280
  - Robotic: cs287
  - NLP: cs288
  - Starcraft competition
  - … and more; ask if you’re interested

That’s It!

- Help us out with some course evaluations
- Have a good break, and always maximize your expected utilities!