Classical RL Setting

• Learner seeks to maximize reward
• Sequential environment (MDP)
Markov Decision Processes

Model of sequential environments (Bellman 57)

- $n$ states, $k$ actions, discount $0 \leq \gamma \leq 1$
- step $t$, agent informed state is $s_t$, chooses $a_t$
- receives payoff $r_t$; expected value is $R(s_t, a_t)$
- probability that next state is $s'$ is $T(s_t, a_t, s')$

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q(s', a')$$

- Optimal behavior is $a_t = \arg\max_a Q(s_t, a)$
- $R$, $T$ unknown; some experimentation needed
Decision Making (Littman & Szepesvari 96)

Define $\otimes_i f(i)$ as a non-expansive summary:

- Maps a vector of values to a scalar.
- $|\otimes_{a'} Q_1(s',a') - \otimes_{a'} Q_2(s',a')| \leq |Q_1 - Q_2|_\infty$
- Ex.: $\otimes_i f(i) = \max_i f(i)$. (min, mean, median.)

Value Iteration converges to the unique solution:

- $Q_{t+1}(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s') \otimes_{a'} Q_t(s',a')$

Q-learning converges to that solution ($\alpha$ decays):

- On $<s,a,r,s'>$ , $Q(s,a) \leftarrow r + \gamma \otimes_{a'} Q(s',a')$

"Simple" matter of learning, exploring, planning.
Multiagent Setting

• Learner seeks to maximize reward
• Sequential environment (stochastic game)
• Environment: Expand action to dual influence.
  \[- R_1(s,(a,b)), R_2(s,(a,b)), T(s,(a,b),s') \]

• Can collide with partner.
  1. **All** first to goal get +100.
  2. Paid only if all reach goal.
Stochastic Games

Model of sequential environments (Shapley 53)<57!

- 2 players, \( n \) states, \( k \) actions per player, \( 0 \leq \gamma \leq 1 \)
- step \( t \), agents informed state is \( s_t \), choose \( a_t, b_t \)
- receive payoff \( r_t^i \); expected value is \( R_i(s_t, (a_t, b_t)) \)
- probability that next state is \( s' \) is \( T(s_t, (a_t, b_t), s') \)

\[
\begin{align*}
\text{agent} & \quad \xrightarrow{S_t} \quad \text{environment: } T, R \quad \xleftarrow{S_t} \quad \text{agent} \\
& \quad \xrightarrow{a_t} \quad r_t^1 \quad \xleftarrow{b_t} \quad r_t^2
\end{align*}
\]

- But, how do we make decisions?
Sequential and Multiagent?

Need something like Bellman equation:

$$Q_1'(s,(a,b)) = R_1(s,(a,b)) + \gamma \sum_{s'} T(s,(a,b),s') \otimes_{(a',b')} Q_1(s',(a',b'))$$

But, how summarize values in $s'$?

Who are you?
Two Convergent Perspectives

Friend-Q  JAL (Claus & Boutilier 98, Littman 01)
⊗(a',b') \( Q_1(s', (a', b')) = \max_{(a', b')} \max Q_1(s', (a', b')) \)
• Other player super helpful
• Easy to compute (joint behavior)

Foe-Q  minimax-Q (Littman & Szepesvári 96)
⊗(a',b') \( Q_1(s', (a', b')) = \min_{(a', b')} \max Q_1(s', (a', b')) \)
• Other player lives to make you suffer
• Somewhat easy to compute (linear program)
Analysis

• Good news:
  – max and minimax are non-expansions.
  – Friend-Q, Foe-Q, VI converge to unique solution.

• Bad news:
  – Except in special situations, unrealistic.
Analysis

- **Good news:**
  - max and minimax are non-expansions.
  - Friend-Q, Foe-Q, VI converge to unique solution.

- **Bad news:**
  - Except in special situations, unrealistic.
Cooperative/Defensive

• More reasonable solution:
  – Pareto optimal
  – Nash / subgame perfect
  – Hard to find?
Hallway

- How do people handle it?

- MTurk: 19 matches; 20 rounds (each max 30 steps)
One Interesting Example

• http://research.clps.brown.edu/mkho/SigRL/Exp1/analysis/visualizer.html#trial/7
What Would You Do?

- **Trust**: Players move out of each other’s way and/or wait to allow both to score.
- **CD** (cooperative/defensive): Players reach goal without any risky moves.
- **Surrender** (friend): One player lets the other score.
- **Alternate** (fair friend): Take turns surrendering.
- **Stalemate** (foe): Players prevent each other from scoring.
- **Compete**: Players are still bumping into each other in the final rounds.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Individual treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compete</td>
<td>32%</td>
</tr>
<tr>
<td>Trust</td>
<td>21%</td>
</tr>
<tr>
<td>Surrender</td>
<td>21%</td>
</tr>
<tr>
<td>CD</td>
<td>11%</td>
</tr>
<tr>
<td>Alternate</td>
<td>11%</td>
</tr>
<tr>
<td>Stalemate</td>
<td>5%</td>
</tr>
</tbody>
</table>
Types of Strategies

Trust

CD

Surrender

Alternate

Stalemate

Compete
Thinking You Are Helping Helps

- Changing payoffs creates a sense of joint intentionality.
- “Trust” easier to find.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Individual treatment</th>
<th>Team treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compete</td>
<td>32%</td>
<td>0%</td>
</tr>
<tr>
<td>Trust</td>
<td>21%</td>
<td>76%</td>
</tr>
<tr>
<td>Surrender</td>
<td>21%</td>
<td>8%</td>
</tr>
<tr>
<td>CD</td>
<td>11%</td>
<td>16%</td>
</tr>
<tr>
<td>Alternate</td>
<td>11%</td>
<td>0%</td>
</tr>
<tr>
<td>Stalemate</td>
<td>5%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Desiderata

• Agents try to work together if possible.
• Agents learn from each other to cooperate.
• Agents can both lead (adopt mutually beneficial behavior) and follow (take cues from other agent).
Some Approaches

• Maximize and act (pure leader)
  – agents can adopt incompatible behaviors

• RL (pure follower)
  – works, but can take almost forever

• Memorization
  – “superstitions”

• Model-based learning
  – not too bad, but weirdly reactive

• Learn “norm” preferences, maximize and act
  – use shared experience as a coordination point
Algorithmic Experiments

Hallway (human comparisons)

Door

Intersection

Long hall

No compromise (no CD strategy)
Greedy Rewards

- By 30,000 rounds, Q-learning figured out how to cooperate.
Subjective Rewards Have Impact

Percent of Learning Attempts Where Agent Found CD Policy

Game (player)

% Learning Attempts

- Hallway
- No Compromise
- Door
- Long Hall (Orange)
- Long Hall (Blue)
- Intersection (Orange)
- Intersection (Blue)
Self Play By Type

- **Hallway**: Selfish (1500) vs. Fair (2400)
- **Intersection**: Selfish (2000) vs. Fair (2200)
- **Door**: Selfish (1200) vs. Fair (1300)
- **Long Hall**: Selfish (2800) vs. Fair (2400)
- **No Compromise**: Selfish (800) vs. Fair (1400)

Average Reward
Norm Learning Algorithm

• Player tries to "explain" (via IRL) choices from prior matches via preference rewards.
• Player creates joint plan to maximize total reward and plays its part.

Algorithm 1 Batch Learning\((\langle I, S, A^I, T, R^I \rangle, \gamma, D, T, B_\Theta)\)

Require: stochastic game \((I, S, A^I, T, R^I)\); discount factor \(\gamma\); multi-agent norm-following demonstrations \(D\); team function \(T\); and parameterized bias function family \(B_\Theta\).

\[
\begin{align*}
A^M &:= \times_i A^I \\
R^M(s, a, s') &:= T(R^I, s, a, s') \\
R^S_\Theta(s, a, s') &:= R^M(s, a, s') + B_\Theta(s, a, s') \\
R^S_\theta &:= \text{IRL}(\langle S, A^M, T, R^S_\Theta \rangle, \gamma, D) \\
\text{return } R^S_\theta
\end{align*}
\]
Learning Results

- The learning results from each of the 5 interactive matches.
- The top image for a match is the first round of interaction of the agents. The bottom image for a match is the learned behavior.
More Human Like

<table>
<thead>
<tr>
<th></th>
<th>Human vs Human</th>
<th>Batch Norm IRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust</td>
<td>76%</td>
<td>72%</td>
</tr>
<tr>
<td>CD</td>
<td>16%</td>
<td>4%</td>
</tr>
<tr>
<td>Stalemate</td>
<td>0%</td>
<td>24%</td>
</tr>
<tr>
<td>Surrender</td>
<td>8%</td>
<td>0%</td>
</tr>
</tbody>
</table>

- 20 rounds.
- Q-learning would still be 100% “compete”.
- Gearing up to play vs. people.
Conclusion

• Standard RL picture: Lonely world, indeed.

• Humans and many animals live in worlds populated by other similar individuals.

• They may have their own reward functions, but they don’t have their own environments.

• Often no right answer: “Unsolvable”.

Multiagent decision making takes place in a “culture”.