Decision Making in Robots and Autonomous Agents

Learning in Repeated Interactions

Stefano V. Albrecht
School of Informatics

27 February 2015

Learning in Repeated Interactions

- How can agent **learn** to interact with other agents?
- What kind of behaviour do we want to learn?
- Learn individually or together?
- Many different methods...
- In this lecture: **reinforcement learning**

Recap

Markov Decision Process:

- states S, actions A
- stochastic transition $P(s'|s, a)$
- utility/reward $u(s, a)$ *(can be random variable)*

Reinforcement Learning:

- "reinforce" good actions
- learn optimal action policy π^*
- e.g. value iteration, policy iteration, ...

  $\rightarrow$ require knowledge of model, e.g. P/u

Q-Learning

What if transition and reward function unknown?

- take action a^t in current state s^t
- only see immediate reward r^{t+1} and next state s^{t+1}

  $\rightarrow$ need **model-free** reinforcement learning

Q-Learning (Watkins & Dayan, 1992)

- store table $Q(s, a)$ for $s \in S, a \in A$
- simple update rule:

$$Q(s^t, a^t) \leftarrow (1 - \alpha) Q(s^t, a^t) + \alpha \left[r^{t+1} + \gamma \max_{a' \in A} Q(s^{t+1}, a') \right]$$

- learns optimal Q-values under certain conditions

Q-Learning in Stochastic Games

Can we use Q-learning for interactive setting?

- general and simple nature appealing
- just learn to interact "on the fly"
- **but:** application not straight-forward, many problems...

  $\rightarrow$ will discuss some problems later

We consider two examples:

- Joint Action Q-Learning (Claus & Boutillier, 1998)
- Nash Q-Learning (Hu & Wellman, 2003)

(Other examples exist)

Joint Action Q-Learning (JAL) (Claus & Boutillier, 1998)

- Assume two players, i and j
- We observe state s^t, actions a_i^t, a_j^t, and results s^{t+1}, r_i^{t+1}
- Store table $Q(s, a_i, a_j)$ where $s \in S, a_i \in A_i, a_j \in A_j$
- Update rule:

$$Q(s^t, a_i^t, a_j^t) \leftarrow (1-\alpha)Q(s^t, a_i^t, a_j^t) + \alpha \left[r_i^{t+1} + \gamma \max_{a_i' \in A} EV(s^{t+1}, a_i') \right]$$

$$EV(s, a_i) = \sum_{a_j \in A_j} P_j(s, a_j) Q(s, a_i, a_j)$$

- $P_j(s, a_j)$ is empirical frequency distribution of j's past actions in state s (**fictitious play**, Brown 1951)

JAL and Nash Equilibrium

- Assume both players controlled by JAL agent
- Assume common payoffs (e.g. players receive same rewards)
- Many other assumptions...

Theorem 1 *Let E_t be a random variable denoting the probability of a (deterministic) equilibrium strategy profile being played at time t. Then for both ILs and JALs, for any $\delta, \varepsilon > 0$, there is an $T(\delta, \varepsilon)$ such that*

$$\Pr(|E_t - 1| < \varepsilon) > 1 - \delta$$

for all $t > T(\delta, \varepsilon)$.

(Claus & Boutillier, 1998)

Nash Q-Learning (NashQ) (Hu & Wellman, 2003)

- Assume two players, i and j
- We observe state s^t, actions a_i^t, a_j^t, and results $s^{t+1}, r_i^{t+1}, r_j^{t+1}$
- Store table $Q(s, a_i, a_j)$ where $s \in S, a_i \in A_i, a_j \in A_j$
- Update rule:

$$Q(s^t, a_i^t, a_j^t) \leftarrow (1-\alpha)Q(s^t, a_i^t, a_j^t) \; + \; \alpha \left[r_i^{t+1} + \gamma NashQ(s^{t+1}) \right]$$

$$NashQ(s) = \sum_{a_i \in A_i} \sum_{a_j \in A_j} \pi_i(s, a_i)\pi_j(s, a_j)Q(s, a_i, a_j)$$

- (π_i, π_j) is (possibly mixed) **Nash equilibrium profile** for matrix game defined by $Q(s, \cdot, \cdot)$

NashQ and Nash Equilibrium

- Assume both players controlled by NashQ agent
- Assume several other restrictions ... including:

Assumption 3 *One of the following conditions holds during learning.*[3]

Condition A. *Every stage game* $(Q_t^1(s),\dots,Q_t^n(s))$*, for all* t *and* s*, has a global optimal point, and agents' payoffs in this equilibrium are used to update their Q-functions.*

Condition B. *Every stage game* $(Q_t^1(s),\dots,Q_t^n(s))$*, for all* t *and* s*, has a saddle point, and agents' payoffs in this equilibrium are used to update their Q-functions.*

(Hu & Wellman, 2003)

- Then the learning converges to a Nash equilibrium

Assumptions in Learning Methods

Different methods may make different assumptions, e.g.

Things that can be "seen":

- JAL: s^t a_i^t a_j^t s^{t+1} r_i^{t+1}
- NashQ: s^t a_i^t a_j^t s^{t+1} r_i^{t+1} r_j^{t+1}

Implicit behavioural assumptions:

- JAL: j plays fixed distribution in each state
- NashQ: j plays Nash equilibrium strategy in each state

Many other types of assumptions about structure of game, behaviour of players, ability to observe, etc.

Assumptions in Learning Methods

Often, method can still be used even if assumptions violated:

- Q-learning assumes stationary transition probabilities

  $\rightarrow$ *Is this true in interactive setting?*
- what happens if assumptions violated?
- **know and understand assumptions!**

Bonus question:

What happens if different methods play against each other?

- e.g. JAL vs NashQ
- (Albrecht & Ramamoorthy, 2012)

Excursion:

Ad Hoc Coordination in Multiagent Systems

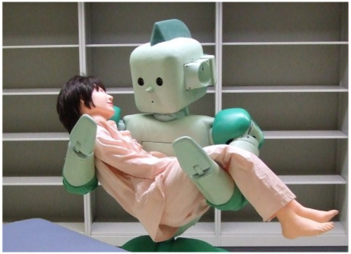
Ad Hoc Coordination

1. You control single agent in system with other agents
2. You and other agents have **goals** (common or conflicting)
3. You want to be **flexible**: other agents may have large variety of behaviours
4. You want to be **efficient**: not much time for learning, trial and error, etc.
5. You don't a priori know how other agents behave

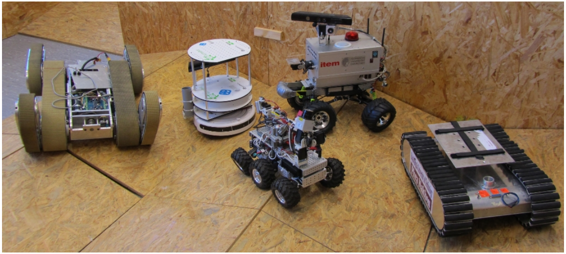
Ad Hoc Coordination

Applications:

- Human-robot interaction
- Robot search and rescue
- Adaptive user interfaces
- Financial markets
- ...



©Biomimetic Control Research Center RIKEN



©Team Hector Darmstadt, Technische Universität Darmstadt

Ad Hoc Coordination

Human-robot interaction:

- Humans can exhibit large variety of behaviours for given task

  $\rightarrow$ need **flexibility**!

- Humans expect machines to learn and react quickly

  $\rightarrow$ need **efficiency**!

- Machine does not know ahead of time how human behaves

  $\rightarrow$ **no prior coordination** of behaviours!

Ad Hoc Coordination

Hard problem:

- Agents may have large variety of behaviours
- Behaviours **initially unknown**

General learning algorithms not suitable:

- Require long learning periods (e.g. RL)
- Often designed for homogeneous setting
- Many restrictive assumptions (discussed earlier)

Idea

Reduce complexity of problem by assuming that:

1. Agents draw their **latent policy** from some set
2. Policy assignment governed by **unknown distribution**

If policy set known:

- Learn distribution, play best-response

If policy set unknown:

- "Guess" policy set, find closest policy, play best-response

Idea

Hypothesise ("guess") Policy Types

Stochastic Bayesian Game

- state space S, initial state $s^0 \in S$, terminal states $\bar{S} \subset S$
- players $N = \{1, ..., n\}$ and for each $i \in N$:
  - set of actions A_i (where $A = \times_i A_i$)
  - **type space** Θ_i (where $\Theta = \times_i \Theta_i$)
  - payoff function $u_i : S \times A \times \Theta_i \to \mathbb{R}$
  - strategy $\pi_i : \mathbb{H} \times A_i \times \Theta_i \to [0, 1]$
    $\mathbb{H}$ is set of histories $H^t = \langle s^0, a^0, ..., s^t \rangle$ s.t. $s^\tau \in S, a^\tau \in A$
- state transition function $T : S \times A \times S \to [0, 1]$
- **type distribution** $\Delta : \Theta \to [0, 1]$

(Albrecht & Ramamoorthy, 2014)

Harsanyi-Bellman Ad Hoc Coordination (HBA)

Canonical formulation **HBA**:

$$a_i^t \sim \arg\max_{a_i \in A_i} E_{s^t}^{a_i}(H^t)$$

where

$$E_s^{a_i}(\hat{H}) = \sum_{\theta_{-i}^* \in \Theta_{-i}^*} \Pr(\theta_{-i}^* | H^t) \sum_{a_{-i} \in A_{-i}} Q_s^{a_{i,-i}}(\hat{H}) \prod_{j \neq i} \pi_j(\hat{H}, a_j, \theta_j^*)$$

$$Q_s^a(\hat{H}) = \sum_{s' \in S} T(s, a, s') \left[u_i(s, a, \alpha) + \gamma \max_{a_i} E_{s'}^{a_i} \left(\langle \hat{H}, a, s' \rangle \right) \right]$$

(Albrecht & Ramamoorthy, 2014)

References (Reading List)

In order of appearance:

C. Watkins, P. Dayan: Q-Learning. MLJ, 1992

C. Claus, C. Boutilier: The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. AAAI, 1998

J. Hu, M. Wellman: Nash Q-Learning for General-Sum Stochastic Games. JMLR, 2003

G. Brown: Iterative Solutions of Games by Fictitious Play. Activity Analysis of Production and Allocation, 1951

S. Albrecht, S. Ramamoorthy: Comparative Evaluation of MAL Algorithms in a Diverse Set of Ad Hoc Team Problems. AAMAS, 2012

S. Albrecht, S. Ramamoorthy: On Convergence and Optimality of Best-Response Learning with Policy Types in Multiagent Systems. UAI, 2014