
# Reinforcement learning for Data Management Scientists

Pierre Senellart

29 March 2014, *NUS*

# Plan

# Reinforcement learning

Reinforcement learning (Sutton and Barto 1998) aims at optimizing the following process:

- An agent interacts with the world by performing a sequence of actions making some transitions
- Each action yields, probabilistically, some reward and some observation
- Available actions may depend on the current state; reward and observation distribution depend on the current state
- The objective is to maximize total reward (within a limited number of actions $N$) or some other monotonic function of the sequence of future rewards

# Reinforcement learning

Reinforcement learning (Sutton and Barto 1998) aims at optimizing the following process:

- An agent interacts with the world by performing a sequence of actions making some transitions
- Each action yields, probabilistically, some reward and some observation
- Available actions may depend on the current state; reward and observation distribution depend on the current state
- The objective is to maximize total reward (within a limited number of actions $N$) or some other monotonic function of the sequence of future rewards

Personal perspective on the topic, from a data management viewpoint

# Formally...

At each time $t$, the agent is in state $X(t)$, has observation $Y(t)$ (correlated with $X(t)$) receives reward $R(t)$, and has just performed action $A(t)$.

Probability distributions for:

state transition: $\Pr(X(t) \mid X(t-1), A(t))$

observation output: $\Pr(Y(t) \mid X(t-1), A(t))$

reward: $\Pr(R(t) \mid X(t-1), A(t))$

These probability distributions are usually assumed to be unknown. $A(t)$ is freely chosen. $R(t)$ and $Y(t)$ are known (once time $t$ is reached). Sometimes one assumes $Y(t) = X(t)$ (MDPs) or that $Y(t)$ and $X(t)$ is correlated with $Y(t)$ in some fashion.

cf. `http://www.cs.ubc.ca/~murphyk/Bayes/pomdp.html`

# Exploration vs Exploitation

Inherent trade-off in reinforcement learning between:

Exploration. Try out many different actions, so as to discover actions that have a high reward, and states from which it is possible to get a high reward.

Exploitation. Focus on the action proven to yield the highest reward so far, to maximize overall reward.

Different strategies, simple ones: (Vermorel and Mohri 2005)

$\epsilon$-greedy: With probability $\epsilon$, explore; otherwise, exploit.

$\epsilon$-decreasing: $\epsilon$-greedy with a decreasing value of $\epsilon$ as $t$ increases.

$\epsilon$-first: First $\epsilon N$ actions are for exploration, remaining for exploitation.

# Connection with Active Learning

- Active learning (Settles 2012): how to optimally use an oracle to label training data to build a learning model (e.g., a classifier)
- Both fields are concerned with answering the question: "What is the next best thing to do?"
- Reinforcement learning assumes an inherent probabilistic model on rewards, observations w.r.t. a state
- In active learning, the reward comes indirectly from the fact that the learning model improves
- Some problems can be modeled in one framework or the other

# Plan

# Multi-armed bandits (MABs)

- Stateless model, $X(t)$ constant
- $k$ actions ($k$-armed bandit), $A(t) \in \{1, \ldots, k\}$
- All actions independent
- Unknown reward distribution $\Pr(R(t) \mid A(t))$
- Maximize, e.g., $\sum_{t=1}^{N} R(k)$
- Exploration (try different arms) vs. exploitation (use best arm so far)
- Distributions of rewards interpolated from sample trials

Heavily researched problem, very simplistic

# Markov Decision Processes (MDP)

- Adding states to multi-armed bandits
- Total observability: $Y(t) = X(t)$
- Reward distribution $\Pr(R(t) \mid A(t), X(t-1))$ and state transition distribution $\Pr(R(t) \mid A(t), X(t-1))$ usually assumed unknown (simplified settings consider one of these known)
- Exploration more complicated: one needs not only try different actions, but also explore the state space. Infeasible if the state space is too large!
- Lots of literature, trade-off exploration/exploitation less well understood than for MABs
- Total observability not realistic in some applications

# Partially Observable MDPs (POMDPs)

- Partial observability: $Y(t) \neq X(t)$ but $Y(t)$ is a probability distribution over possible $X(t)$'s
- Corresponds to the situation where one maintains an uncertain current state of the world (cf. probabilistic databases)
- Reward distribution $\Pr(R(t) \mid A(t), X(t-1))$ and state transition distribution $\Pr(R(t) \mid A(t), X(t-1))$ usually assumed unknown
- Exploration even more complicated by the fact that the current state is uncertain, meaning the feedback of action trials does not provide an easy data point to interpolate reward and state transition distributions
- Good model for many real-world applications

# Plan

# Bandits for Focused Crawling (Gouriten, Maniu, and Senellart 2014)

- Focused crawling: crawl a graph (e.g., the Web, a social network) focusing on nodes that have a good score (e.g., w.r.t. a keyword query)
- Score of a node unknown until it has been crawled
- One wants to predict the node with highest score to know what the next best node to crawl is
- Many possible predictors: score of the parent node, average score of other nodes pointed by the parent node, depth in the crawl from seed nodes, topic-sensitive PageRank computed online, etc.
- The best predictors will depend on the graph crawled
- Use MABs to model the problem: predictors are arms, reward is high if predicted score close to actual score; use MAB strategies to solve it

# MDPs for Data Cleaning (Benedikt, Bohannon, and Bruns 2006)

- Incomplete or low-quality data
- Sources (experts, Web services, etc.) can be queried (for a cost) to get more precise data
- Specific goal (e.g., a query)
- How to estimate the value of accessing a source?
- Modeling as a MDP:
  - States are the history of all source accesses
  - Actions are source accesses or cleaning decisions
  - Rewards are a negative function of the cost to access a source, and a positive function of the gain towards the goal

# POMDPs for Intensional Data Querying (Amarilli and Senellart 2014)

- Lots of data cannot be accessed in extension, but intensionally (for a cost): crowdsourcing, Web services, costly annotation tools, etc.
- Data sources are uncertain
- We maintain a current knowledge of the world, which is a probabilistic database; in other words, a probability distribution over possible MDP states
- Again, rewards are a negative function of the cost and positive functions of the gain for a query

# Plan

# State Size Explosion

- Existing algorithms for solving MDPs do not scale to huge state space
- In data management applications, the state (at least) contains the whole database, and sometimes even exponentially many states w.r.t. the size of the data
- Possible to reduce the state space, e.g., compute the MDP obtained by quotienting the state by some equivalence relation (Givan, Dean, and Greig 2003)

# Structure of States and Logic Constraints

- In the MDP world, states are abstract
- In the database world, states are very structured (e.g., each state corresponds to a structured database)
- Often, additional constraints (e.g., keys)
- Some early works dealing with MDPs where states have an inherent structure (Otterlo 2009)

# Delayed Rewards

- Reinforcement learning postulates that every action has an immediate reward
- In database world, the reward is typically delayed (reward when the query has been answered)
- Lots of states/actions without immediate rewards: Messes up the exploration/exploitation trade-off

# Heterogeneous Cost/Reward

- Reward is unidimensional
- In database applications, there are various costs/rewards:
  - The cost of performing an action (computational cost, budget for accessing the crowd, network bandwidth, I/O, etc.)
  - The reward of getting a query answer
  - ... and the computational cost of choosing the next best thing to do

# Plan

# Conclusion

- Lots to learn from the machine learning and artificial intelligence communities
- Main difference for data management problems: the current state is a huge, structured, database; the goal is a complex logical constraint on the data
- To go further: two textbooks (Sutton and Barto 1998; Puteman 2005)

# Conclusion

- Lots to learn from the machine learning and artificial intelligence communities
- Main difference for data management problems: the current state is a huge, structured, database; the goal is a complex logical constraint on the data
- To go further: two textbooks (Sutton and Barto 1998; Puteman 2005)

Merci.

Amarilli, A. and P. Senellart (Feb. 2014). *UnSAID: Uncertainty and Structure in the Access to Intensional Data*. Preprint available at http://pierre.senellart.com/publications/amarilli2014unsaid.

Audibert, J.-Y., R. Munos, and C. Szepesvári (2009). "Exploration-exploitation tradeoff using variance estimates in multi-armed bandits". In: *Theor. Comput. Sci.* 410.19.

Benedikt, M., P. Bohannon, and G. Bruns (2006). "Data Cleaning for Decision Support". In: *CleanDB*.

Givan, R., T. L. Dean, and M. Greig (2003). "Equivalence notions and model minimization in Markov decision processes". In: *Artif. Intell.* 147.1-2.

Gouriten, G., S. Maniu, and P. Senellart (Mar. 2014). *Scalable, Generic, and Adaptive Systems for Focused Crawling*. Preprint available at http://pierre.senellart.com/publications/gouriten2014scalable.

Otterlo, M. van (2009). *The Logic of Adaptive Behavior - Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains*. Vol. 192. Frontiers in Artificial Intelligence and Applications. IOS Press.

Puteman, M. L. (2005). *Markov Decision Processes. Discrete Stochastic Dynamic Programming*. Wiley.

Settles, B. (2012). *Active Learning*. Morgan & Claypool Publishers.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning*. MIT Press.

Vermorel, J. and M. Mohri (2005). "Multi-armed Bandit Algorithms and Empirical Evaluation". In: *ECML*.