Bachelor's Thesis Proposal
Evaluation of Exploration Algorithms in Reinforcement Learning

Alexander Böttcher

Bachelor Student of Computer Science at
Hochschule Furtwangen University

1. Problem description

The aim of my Bachelor Thesis is to provide an evaluation of a broad range of exploration algorithms in a discrete version of the BRIO labyrinth environment. Based on these results, a proposal will be given that suggests an algorithm for an agent acting in the real BRIO labyrinth.

2. Introduction

Reinforcement Learning is a kind of learning that is based on the interaction of agents with the environment. An agent performs actions in an environment and the environment will afterwards provide the agent with a reward and the next state. The goal of the agent is to maximize the discounted longterm reward by repeatedly choosing actions. At the beginning, agents usually have none or only limited knowledge of the environment and they must gain knowledge by interaction with the environment. The main issue in Reinforcement Learning is the trade-off between further exploration of the environment or exploitation of the already available knowledge. While exploitation maximizes the reward based on the current knowledge exploration can lead to a higher long term reward by selecting actions which appear not optimal based on the current knowledge. Algorithms handling the exploration exploitation trade-off are called exploration algorithms.

Most Reinforcement Learning (RL) research is based on Markov Decision Processes (MDP). This special class of Reinforcement Learning tasks includes only state signals which have the Markov property. A state signal has the Markov property, if the state signal compactly summarizes the past sensations that are needed to perform action selection. Because of the Markov property of the states, decisions and values are a function only of the current state. The function that maps a state to an action is called a policy. In order to maximize the reward the agent has to learn an optimal policy. It is common in RL that the agent starts with an arbitrary initialized policy that is improved as new knowledge of the environment is experienced by exploration.

There are four basic exploration algorithms. 1) greedy: a greedy algorithm is an algorithm that makes the locally optimal choice at each stage. With respect to the value estimates, the greedy exploration algorithm will always select the action with the highest value estimate. 2) epsilon greedy: selects actions most of the time greedy and only a small fraction ε of the decisions are actions chosen randomly from all available actions. 3) softmax: the probability distribution of the actions to be chosen is based on the distribution of the estimated action-values and also depends on a temperature τ to make the distribution more greedy or equiprobable. 4) pursuit methods: the probability to choose the greedy action, based on the current action-value estimates, is increased after every time step. [1]

There are more sophisticated algorithms and the research area is still very active. The next section will give an overview of recent results of research.

The theoretical statements about exploration algorithms are the boundaries of space, computational and sample complexity. But sample complexity boundaries will not answer which of the algorithms has the highest learning speed in a certain environment. The algorithm's learning speed varies depending on the environment. Which algorithm fits best in a certain environment needs to be empirically evaluated or

mathematically proven for the specific environment.

My work focuses on an environment which is a discrete version of the BRIO environment.

3. State-of-the-art Exploration Algorithms

This section lists a selection of current approaches to handle the exploration exploitation trade-off. There is a wide range of approaches and algorithms available and the following list gives an overview of them.

E^3

E^3 is a model-based algorithm that is proven to be a "Probably Approximately Correct" MDP (PAC-MDP) [2]. Theoretical work normally guarantees that a certain exploration algorithm finds an optimal policy within an infinite number of time steps, PAC-MDPs on the other hand are proven to learn a near-optimal policy in a polynomial time and experience. The name E^3 is an abbreviation for "explicit explore or exploit". The algorithm collects data and builds a model of the environment. States become known after they have been visited a certain number of time. There are two policies that do exploitation or exploration respectively. Every time the agent arrives in an known state during exploration or after a certain amount of steps during exploitation, the agent does off line computations and chooses either to continue with exploration or exploitation. The decision is based on the accuracy of the exploitation policy. [3]

R-Max

R-Max is similar to E^3. It is also a model-based PAC-MDP and also maintains a set of known states which is defined in the same way as in E^3. But the exploration exploitation trade-off is made differently. R-Max uses initially optimistic parameters that assumes all states and actions yield maximum reward. The agent starts with an empty model. Whenever a state was visited often enough it is added to the set of known states and the information about the state is added to the model. After each model update the policy will be recomputed to be the optimal policy of the updated model. The exploration is given by initial optimistic values. [4]

Optimistic initial model (OIM)

The optimistic initial model algorithm combines several approaches in one algorithm. It is model-based and it uses optimistic initial values as R-Max does. The algorithm uses an improved prioritized sweep algorithm as an asynchronous dynamic programming algorithm. In order to save the initial boost from being swept away two value estimates are included in the model: The first is a value estimate for the reward and the second is a value estimate for the exploration. So the updates of the prioritized sweeping distributes the reward estimate and the exploration estimate separately. The actions are selected greedily with respect to the reward estimate plus the exploration estimate bonus. The bonus decreases as the number of visits increases. [5]

Topological Q-Learning, Topological R-Max

Topological Q-Learning (TQL) is divided into two parts. The first part is standard Q-Learning, but additionally to the Q-values a directed graph is build on the gained experience with the environment representing the MDP. In the second part the approximated graph is separated in its strongly coupled components (SCC). These SCCs build a directed acyclic graph (DAG) that has a topological order. Trials are run on each component separately. The trails are run on a certain component until the estimated values of the respective component converged. Please see the paper for details and a description of Topological R-Max. [6]

Delayed Q-learning

This algorithm is similar to Q-Learning. Updates of the Q-values are delayed and also some criteria must hold in order to update the values at all. The main criterion is that the difference between the old estimate and the new must be greater than a certain value. The updates include a exploration bonus and the action selection is greedy with respect to the estimated values. Exploration is done by adding the exploration bonus to the value update. The algorithm is model free and proven to be a PAC-MDP. [7]

MBIE MBIE-EB

Interval estimation methods use a confidence interval for the mean of each action's estimated value. The action selection chooses the action with the highest upper tail of the confidence interval. If the interval is big, the mean return may be loose. While the experience grows the intervals shrink and the accuracy rises. How big the confidence intervals are and how they are computed is explained in [8].

Experts Algorithms

Experts algorithms is a very different approach. It is based on experts. An expert is a particular strategy recommending actions. An experts algorithm combines the recommendations of several given "experts" into another strategy of choosing actions. It can be shown that expert algorithms can perform in the long-run as well as the best expert. There are two phases. The exploration phase selects a random expert for a certain amount of steps. The exploitation phase uses the expert which showed best results in the past. The selection is done in an epsilon-greedy way. Most of the time the exploitation phase is chosen and for a small fraction of the time the exploration phase is chosen. [9]

Further Approaches

There are more algorithms to consider, most notable: Hierarchical Reinforcement Learning MAXQ-Q [10] and Reinforcement Learning with Decision Trees RL-DT [11].

4. Project Definition

The project is divided into the following parts:

Research

This part includes searching and reading of current research papers. Additionally I have to check the relevance of the respective work. Based on that I decide whether or not to include it into the evaluation. The selection is necessary, because there are too many papers with different approaches and modifications available to include them all.

Implementation

The implementation covers changes to the Maja Machine Learning Framework as well as the implementation of the selected algorithms and the discrete Brio environment.

Evaluation

Before the evaluation can be done, the test scenario must be planed. Questions that need to be answered before the evaluation can start:

- How many different Brio boards should be used?
- What should the boards look like?
- How many runs should be done?
- How many episodes should a run consist of?
- Which method should be used to optimize the parameters of the algorithms?

The last step of the evaluation is to interpret the results and present them in an intuitive way.

Write the Thesis

I will write the main parts of the thesis in parallel to the steps mentioned above. At the end I scheduled time to focus on the writing only. The work left includes checking the whole document and make it coherent.

5. Evaluation method, used criteria

The evaluation is done empirically. Each of the selected algorithms will be run several times for a certain amount of episodes. Several runs are necessary, since a single run is not accurate, because of the impact of randomness on the results. So the averaged results of the algorithms are compared.

The criterion for the evaluation is the learning speed. I will neither evaluate the memory space needed by the algorithms nor the computational time needed. The learning speed will be assessed in two ways:

- the accumulated reward after n episodes and
- the mean reward of a policy that has been trained for n episodes.

6. Schedule - Work plan

The following table gives a rough estimate of the next steps and the time scheduled to work on them.

#	Task name	duration
1	Further research	1 month
2	Implementation	$\frac{3}{2}$ months
3	Evaluation	1 month
4	Complete writing the Thesis	$\frac{1}{2}$ month

The submission deadline for the thesis is August 31st, 2009.

References

1. Sutton, Richard S. and Barto, Andrew G., *Introduction to Reinforcement Learning*, MIT Press, Cambridge, MA, USA (1998).
2. Kakade, S. M., *On the sample complexity of reinforcement learning. (Doctoral dissertation)*, Gatsby Computational Neuroscience Unit, University College London (2003).
3. Kearns, Michael and Singh, Satinder, "Near-Optimal Reinforcement Learning in Polynomial Time," *Mach. Learn.*, 49, pp. 209-232, Kluwer Academic Publishers, Hingham, MA, USA (2002).
4. Brafman, Ronen I., "R-max - a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, 3, pp. 213-231 (2002).
5. Szita, Istvan and Lincz, Andras, "The many faces of optimism: a unifying approach" in *ICML '08: Proceedings of the 25th international conference on Machine learning*, pp. 1048-1055, ACM, New York, NY, USA (2008).
6. Dai, Peng and Strehl, Alexander L. and Goldsmith, Judy, "Expediting RL by Using Graphical Structures (Short Paper)" in *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 1325-1328, INESC, Lisbon, Portugal, Portugal (2008).
7. Strehl, Alexander L. and Li, Lihong and Wiewiora, Eric and Langford, John and Littman, Michael L., "PAC model-free reinforcement learning" in *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pp. 881-888, ACM, New York, NY, USA (2006).
8. Strehl, Alexander L. and Littman, Michael L., "An analysis of model-based Interval Estimation for Markov Decision Processes," *J. Comput. Syst. Sci.*, 74, pp. 1309-1331, Academic Press, Inc., Orlando, FL, USA (2008).
9. de Farias, D. and Megiddo, N., "Exploration-exploitation tradeoffs for experts algorithms in reactive environments" in *Advances in neural information processing systems 17.*, pp. 409-416, MIT Press, Cambridge, MA, USA (2005).
10. Dietterich, Thomas G., "An overview of MAXQ hierarchical reinforcement learning" in *SARA '02: Proceedings of the 4th International Symposium on Abstraction, Reformulation, and Approximation*, pp. 26-44, Springer Verlag, London, UK (2000).
11. Hester, Todd and Stone, Peter, "Generalized Model Learning for Reinforcement Learning in Factored Domains" in *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Budapest, Hungary (2009).