Formatting ICML-99 Submissions*

Sašo Džeroski[1], Luc De Raedt[2], Hendrik Blockeel[2]

[1] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium

Abstract

This is a fictituos submission obeying ICML-99 formatting requirements. It re-uses text from the ICML-98 paper on relational reinforcement learning by the above authors. ...

RRL is presented, a learning technique that combines reinforcement learning with relational learning or inductive logic programming. Due to the use of a more expressive representation language to represent states, actions and Q-functions, relational reinforcement learning can be potentially applied to a new range of learning tasks. One such task that we investigate is planning in the block's world, where it is assumed that the effects of the actions are unknown to the agent and the agent has to learn a policy. In particular, relational reinforcement learning allows to employ structural representations, to make abstraction of specific goals pursued and to exploit the results of previous learning phases when addressing new (more complex) situations.

Keywords: *reinforcement learning, relational learning, inductive logic programming, planning.*

Email address of contact author: Saso.Dzeroski@ijs.si

Phone number of contact author: ++ 386 61 177 3217

Electronic version: gzipped PostScript emailed to icml99@ijs.si

*This paper is submitted jointly to ICML-99 and ILP-99.

1 Introduction

Within the field of machine learning, both reinforcement learning [?] and inductive logic programming (or relational learning) [?, ?] have received a lot of attention since the early nineties. It is therefore no surprise that both Leslie Pack Kaelbling and Rich Sutton (in their invited talks at IJCAI-97, Nagoya, Japan) suggested to study the combination of these two fields.

From the reinforcement learning point of view, this could significantly extend the application perspective. So far, reinforcement learning has by large employed the classical feature vector or attribute value representations. This representation is inadequate for describing certain tasks (such as e.g. the simple block's world). Furthermore, due to the use of variables in relational representations, it is possible to make abstractions of some specific details of the learning tasks, such as the goal pursued. Indeed, when learning to plan in the block's world, one would expect that the results of learning how to stack block a onto block b would be similar to stacking c onto d. Current approaches to reinforcement learning have to retrain from scratch if the goal is changed in this manner. Using relational reinforcement learning retraining is unnecessary. Relational reinforcement learning also allows to exploit the results of one learning task in a simple world when learning in a more complex domain (e.g. going from 3 blocks to 4 blocks in the block's world).

From the inductive logic programming point of view, it is important to address domains such as reinforcement learning. So far, inductive logic programming has mainly studied concept-learning, and largely ignored the rest of machine learning. By demonstrating the potential of relational representations for reinforcement learning, we hope to show that the relational learning methodology does not only apply to concept-learning but to the whole field of machine learning.

...

This paper is organised as follows. In section 2, we view planning (under uncertainty) as a reinforcement learning task, and in section 3, we briefly review reinforcement and in particular Q-learning. Section 4 introduces relational reinforcement learning that combines Q-learning and logical regression trees. In section 5, we present some experiments, and finally, in section 6, we conclude and touch upon related work.

...

2 Discussion

We have presented an approach to planning with incomplete knowledge that combines reinforcement learning and relational regression into a technique called relational reinforcement learning. The advantages of this approach include the ability to use structured representations, which enables us to also describe infinite worlds, and the ability to use variables, which allows to abstract away from specific details of the situations (such as, e.g. the goal). The ability to use results of simpler tasks to bootstrap learning in more complex tasks is also an advantage worth mentioning. Finally, it is easy to incorporate nondeterministic actions within the proposed approach.

...

Relational representations are commonly used in planning approaches. There have also been some attempts to combine planning with relational learning within those appproaches, e.g., within the PRODIGY approach [**?**]. Our approach is related to them through the use of a relational representation. However, it seems that the combination of planning, reinforcement learning and relational learning has not been addressed so far.

The current implementation of TILDE-RT is - according to reinforcement standards - not optimal. The reason is that it is not incremental and also that it may assign different Q values to the same examples at different points in time. This contrasts with Chapman's and Kaelbling's decision tree algorithm that was specifically designed for reinforcement learning [**?**]. We want to stress however that it would be easy (and just a matter of implementation) to develop a first order regression tree algorithm combining the representations of TILDE-RT with the algorithm and performance measures of the approach by Chapman and Kaelbling. Such an integrated approach, which is currently under development, would not suffer from the abovementioned problems.

References

[1] Blockeel, H., and De Raedt, L. (1997) Experiments with Top-down Induction of Logical Decision Trees. Technical Report CW 247, Department of Computer Science, Katholieke Uneversitet Leuven, Belgium. Submitted to *Artificial Intelligence.*

[2] Borrajo, D., and Veloso, M. (1997) Lazy incremental learning of control knowledge for efficiently obtaining quality plans. *AI Review*, 11(1-5): 371–405.

[3] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) *Classification and Regression Trees.* Wadsworth, Belmont.

[4] Blockeel, H., and De Raedt, L. (1997) Lookahead and discretization in ILP. In *Proc. 7th Intl. Workshop on Inductive Logic Programming*, pages 77–84, Springer, Berlin.

[5] Chapman, D., and Kaelbling, L. (1991) Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. In *Proc. 12th Intl. Joint Conf. on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA.

[6] De Raedt, L., and Blockeel, H. (1997) Using logical decision trees for clustering. In *Proc. 7th Intl. Workshop on Inductive Logic Programming*, pages 133–141, Springer, Berlin.

[7] Kaelbling, L., Littman, M., and Moore, A. (1996) Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4: 237–285.

[8] Kramer, S. (1996) Structural regression trees. In *Proc. 13th Natl. Conf. on Artificial Intelligence.* AAAI Press, Menlo Park, CA.

[9] Lavrač, N. and Džeroski, S. (1994) *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, Chichester.

[10] Mitchell, T. (1997) *Machine Learning.* McGraw-Hill, New York.

[11] Muggleton, S., and De Raedt, L. (1994) Inductive logic programming : Theory and methods. *Journal of Logic Programming* 19/20: 629–679.

[12] Tesauro, G. (1995) Temporal difference learning and TD-GAMMON. *Communications of the ACM*, 38(3): 58–68.

[13] Watkins, C., and Dayan, P. (1992) Q-learning. *Machine Learning*, 8: 279–292.