

A Learning Classifier System Approach to Relational Reinforcement Learning

Drew Mellor

B. Comp. Sci. (Hons)

Submitted in partial fulfilment of the requirements for the degree of
DOCTOR OF PHILOSOPHY (COMPUTER SCIENCE)

School of Electrical Engineering and Computer Science

The University of Newcastle

Callaghan, 2308

Australia

February 2008

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

(Signed):

Acknowledgements

A PhD thesis is a sizeable undertaking and invariably depends on contributions from many people besides the author. I would like to extend a warm thank-you to the people in the following list, all of whom, although some of them may not realise it, contributed to this research. Mirka Miller encouraged me to do a PhD in the first place and was instrumental during the application process. Sašo Džeroski gave a keynote talk at ICML 2002 which inspired the thesis topic. My supervisors Stephan Chalup and Huilin Ye gave me their trust and a free hand to develop the topic. Frans Henskens went beyond his professional duty to help secure much needed travel funding for the presentation of my work. Stewart Wilson, Tim Kovacs and Martin Butz provided encouragement and validation at a crucial time. Sašo Džeroski, Kurt Driessens, Martijn van Otterlo and Federico Divina helpfully answered my queries about relational reinforcement learning and other topics. Robert King advised on appropriate statistical tests, while Aaron Scott, David Montgomery and Geoff Martin tirelessly provided valuable technical support without which this thesis would have come to nothing. During the writing phase many people proof read chapters: Alyssa Brugman, Dirk Brugman, Elena Prieto, Erol Engin, Linda Seymour, Michael Quinlan and of course my supervisors, Stephan and Huilin. Throughout the course of the candidature my parents Rob and Cherie Mellor always gave me their support, while Helen Giggins and the legged Robocup team, Michael, Craig, Naomi, Kenny, Steve and the others were at hand for much needed regular diversions. I would also like to thank anyone else that belongs on this list but whom I might have forgotten to add, for which I sincerely apologise. Last but not least, I would like to mention my long time pet cat and companion, “Crackles”, who passed away during the candidature and to whom this thesis is dedicated.

Contents

List of Symbols	vii
Abstract	xiii
1 Introduction	1
1.1 Motivation	2
1.1.1 Blocks World	3
1.1.2 The Reinforcement Learning Problem	4
1.1.3 Structural Regularity	6
1.1.4 First-Order Logic	7
1.2 Approach	9
1.2.1 Learning Classifier Systems	10
1.2.2 Inductive Logic Programming	10
1.2.3 Strengths and Weaknesses	12
1.3 Thesis Objectives	13
1.4 Thesis Outline	14

2	Background	17
2.1	Markov Decision Processes	18
2.1.1	Acting Optimally	20
2.1.2	Solving Markov Decision Processes	22
2.2	Reinforcement Learning	26
2.2.1	The TD(0) Algorithm	26
2.2.2	The Q-Learning Algorithm	28
2.3	Generalisation	30
2.3.1	Function Approximation	31
2.3.2	Aggregation	33
2.3.3	Remarks	35
2.4	Relational Reinforcement Learning	36
2.4.1	Propositionally Factored MDPs	36
2.4.2	Relational Markov Decision Processes	40
2.4.3	Aggregating States and Actions Under an RMDP	43
2.5	Summary	44
3	A Survey of Relational Reinforcement Learning	47
3.1	Connections to Other Fields	48
3.2	Existing RRL Methods and Approaches	49
3.2.1	Static Generalisation	50
3.2.2	Dynamic Generalisation	53

3.2.3	Policy Learning	55
3.2.4	Policy Driven Approaches	58
3.2.5	Other Dynamic Methods	60
3.2.6	Extensions and Related Methods	61
3.3	Dimensions of RRL	64
3.4	Discussion	66
3.5	Summary	69
4	The XCS Learning Classifier System	71
4.1	The XCS System	72
4.1.1	System Architecture	73
4.1.2	The Rule Base	74
4.1.3	The Production Subsystem	77
4.1.4	The Credit Assignment Subsystem	80
4.1.5	The Rule Discovery Subsystem	86
4.2	Accuracy-Based Fitness	91
4.3	Biases within XCS	94
4.3.1	The Generality and Optimality Hypotheses	94
4.3.2	Butz's Evolutionary Pressures	96
4.3.3	Discussion	97
4.4	Alternative Rule Languages	99
4.5	Genetic Algorithms	104

4.6	Summary	106
5	The FOXCS System	109
5.1	Overview	110
5.2	Representational Aspects	112
5.2.1	Background Knowledge	113
5.2.2	Representation of the Inputs	114
5.2.3	Representation of the Rules	115
5.2.4	Expressing Generalisations Within a Single Rule	116
5.3	The Matching Operation	118
5.3.1	The Order of Atoms Within a Rule	119
5.3.2	The Use of Inequations	119
5.3.3	Caching	120
5.4	The Production Subsystem	121
5.5	The Rule Discovery Subsystem	124
5.5.1	Declaring the Rule Language	124
5.5.2	The Covering Operation	127
5.5.3	The Mutation Operations	130
5.5.4	Subsumption Deletion	140
5.6	Implementation Notes	143
5.7	Summary	144
6	Application to Inductive Logic Programming	145

6.1	Experimental Setup	146
6.1.1	Materials	146
6.1.2	Methodology	148
6.2	Comparison to ILP Algorithms	149
6.3	Verifying the Generality Hypothesis	152
6.4	The Effect of Subsumption Deletion on Efficiency	156
6.5	The Effect of Learning Rate Annealing on Performance	157
6.6	The Influence of the Selection Method	162
6.7	Summary	167
7	Application to Relational Reinforcement Learning	171
7.1	Experiments In Blocks World	172
7.2	Scaling Up	181
7.2.1	P-Learning	183
7.2.2	An Implementation of P-Learning for FOXCS	184
7.2.3	Experiments	186
7.3	Summary	192
8	Conclusion	193
8.1	Summary	193
8.2	Contributions	196
8.3	Significance	198
8.4	Future Work	201

A	First-Order Logic	203
A.1	Syntax	204
A.2	Semantics	208
A.3	Herbrand Interpretations	213
A.4	Inference	215
A.5	Induction	216
A.6	Substitution	219
B	Inductive Logic Programming	222
B.1	Learning from Entailment	224
B.2	Learning from Interpretations	226
B.3	Discussion	229
C	The Inductive Logic Programming Tasks	231
C.1	The Prediction of Mutagenic Activity	231
C.2	The Prediction of Biodegradability	234
C.3	Predicting Traffic Congestion and Accidents	235
C.4	Classifying Hands of Poker	238
D	Validity Tests for the Blocks World Environment	241

List of Symbols

The lists below contain symbols and acronyms used commonly throughout the thesis. Where relevant, page numbers are given to where the symbol or abbreviation is introduced.

Logic

\wedge	conjunction
\vee	disjunction
\neg	negation
\leftarrow	implication
\exists	existential quantifier
\forall	universal quantifier
\models	entailment, 215
θ	a substitution
θ^{-1}	an inverse substitution
$const(\Phi)$	the set of constant symbols occurring in the logical sentence Φ
$vars(\Phi)$	the set of variables occurring in the logical sentence Φ

Reinforcement Learning

t	discrete time step, 18
s	a state, 18
a	an action, 18
r	a reward, 18
\mathcal{S}	state space, 18
\mathcal{A}	action space, 18
$T(s, a, s')$	the probability of transition from s to s' under a , 19
$R(s, a)$	expected reward from taking a in s , 19
$\mathcal{A}(s)$	set of actions possible in state s , 19
π	policy, 21
π^*	optimal policy, 22
$\pi(s)$	the action to take in state s under policy π , 21
$V^\pi(s)$	the value of state s under policy π , 21
$V^*(s)$	the value of state s under an optimal policy, 22
$Q^*(s, a)$	the value of state-action pair (s, a) under an optimal policy, 28
γ	discount factor, 21
α	learning rate, 27
ϵ	probability of selecting a random action under an ϵ -greedy policy, 29

Relational Reinforcement Learning

\mathcal{L}	an alphabet over first-order logic
\mathcal{C}	set of constant symbols
\mathcal{F}	set of function symbols
\mathcal{P}	set of predicate symbols
\mathcal{P}_A	set of predicate symbols for representing actions
\mathcal{P}_S	set of predicate symbols for representing states
\mathcal{P}_B	set of predicate symbols for representing background knowledge

Learning Classifier Systems

The following list gives the parameters associated with an individual rule j in XCS (the parameter's name is given in parentheses):

A_j	action advocated by j (action), 74
C_j	the set of states to which j applies (condition), 74
p_j	an estimate of the mean expected payoff for j (prediction), 75
ε_j	an estimate of the mean expected absolute difference between p_j and the actual payoff (error), 75
F_j	determines j 's probability of selection for reproduction (fitness), 75
exp_j	the number of times j has been a member of an action set (experience), 75
ns_j	an estimate of the mean size of the action sets that j has been a member of (niche size), 76

ts_j	the time step when the GA was most recently invoked on an action set that j belonged to (time step), 76
n_j	the number of micro-rules represented by j (numerosity), 76
dv_j	probability that j is deleted (deletion vote), 90

Other important symbols and parameters for XCS:

$[P]$	the current set of rules contained within the system (population), 74
$[M]$	the set of rules matching the current state (match set), 77
$[A]$	the subset of $[M]$ advocating the selected action (action set), 80
$[A]_{-1}$	$[A]$ from the previous time step (action set), 80
N	the maximum number of rules in terms of micro-rules that may exist in $[P]$ at any one time, 77
ρ	the target value when updating p , 81
κ	an inverse measure of ε used for calculating fitness (accuracy), 82
a, b, ϵ_0	parameters for calculating κ , 82
α	learning rate for p , ε , F , and ns updates, 81
γ	discount factor, 81
θ_{GA}	a threshold used by the triggering mechanism for the GA, 89
θ_{del}	a threshold used by the rule deletion mechanism, 90
δ	the fraction of the mean fitness below which a rule's probability of deletion is increased, 90
θ_{sub}	a threshold used by subsumption deletion, 91

The following symbols are specific to FOXCS:

- Φ_j the logical part of j , replacing A_j and C_j , consisting of a definite clause over first-order logic, 115
- μ_i determines the probability of selecting an evolutionary operation, $i \in \{\text{DEL}, \text{C2V}, \text{V2A}, \text{ADD}, \text{V2C}, \text{A2V}, \text{REP}\}$, 134

Acronyms

The following acronyms are used throughout this thesis:

- GA Genetic algorithm, 104
- ILP Inductive logic programming, 10, 222
- LCS Learning classifier system, 10, 71
- MDP Markov decision process, 18
- RRL Relational reinforcement learning, 36, 47

Abstract

Machine learning methods usually represent knowledge and hypotheses using attribute-value languages, principally because of their simplicity and demonstrated utility over a broad variety of problems. However, attribute-value languages have limited expressive power and for some problems the target function can only be expressed as an exhaustive conjunction of specific cases. Such problems are handled better with inductive logic programming (ILP) or relational reinforcement learning (RRL), which employ more expressive languages, typically languages over first-order logic. Methods developed within these fields generally extend upon attribute-value algorithms; however, many attribute-value algorithms that are potentially viable for RRL, the younger of the two fields, remain to be extended.

This thesis investigates an approach to RRL derived from the learning classifier system XCS. In brief, the new system, FOXCS, generates, evaluates, and evolves a population of “condition-action” rules that are definite clauses over first-order logic. The rules are typically comprehensible enough to be understood by humans and can be inspected to determine the acquired principles. Key properties of FOXCS, which are inherited from XCS, are that it is general (applies to arbitrary Markov decision processes), model-free (rewards and state transitions are “black box” functions), and “tabula rasa” (the initial policy can be unspecified). Furthermore, in contrast to decision tree learning, its rule-based approach is ideal for *incrementally* learning expressions over first-order logic, a valuable characteristic for an RRL system.

Perhaps the most novel aspect of FOXCS is its inductive component, which synthesizes evolutionary computation and first-order logic refinement for incremental learning. New evolutionary operators were developed because previous combinations of evolutionary computation and first-order logic were non-incremental. The effectiveness of the inductive component was empirically demonstrated by benchmarking on ILP tasks, which found that FOXCS produced hypotheses of comparable accuracy to several well-known ILP algorithms. Further benchmarking on RRL tasks found that the optimality of the policies learnt were at least comparable to those of existing RRL systems. Finally, a significant advantage of its use of variables in rules was demonstrated: unlike RRL systems that did not use variables, FOXCS, with appropriate extensions, learnt scalable policies that were genuinely independent of the dimensionality of the task environment.