
Preface

Classifier systems are an intriguing approach to a broad range of machine learning problems, based on automated generation and evaluation of condition/action rules. In reinforcement learning tasks they simultaneously address the two major problems of learning a policy and generalising over it (and related objects, such as value functions). Despite over 20 years of research, however, classifier systems have met with mixed success, for reasons which were often unclear. Finally, in 1995 Stewart Wilson claimed a long-awaited breakthrough with his XCS system, which differs from earlier classifier systems in a number of respects, the most significant of which is the way in which it calculates the value of rules for use by the rule generation system. Specifically, XCS (like most classifier systems) employs a genetic algorithm for rule generation, and the way in which it calculates rule fitness differs from earlier systems. Wilson described XCS as an accuracy-based classifier system and earlier systems as strength-based. The two differ in that in strength-based systems the fitness of a rule is proportional to the return (reward/payoff) it receives, whereas in XCS it is a function of the accuracy with which return is predicted. The difference is thus one of credit assignment, that is, of how a rule's contribution to the system's performance is estimated. XCS is a Q-learning system; in fact, it is a proper generalisation of tabular Q-learning, in which rules aggregate states and actions. In XCS, as in other Q-learners, Q-values are used to weight action selection. In rule generation, a rule's weight is an inverse function of the error of its Q-value; thus XCS finds rules which minimise errors in the Q-function, and is able to accurately approximate it. Wilson argued XCS's superiority over earlier systems, and demonstrated improved performance on two tasks. However, the theoretical basis of XCS's results – and indeed those of earlier systems – needed development. Consequently, this thesis has been devoted to the development of theory to explain how and why classifier systems behave as they do. In particular, a theory underlying the difference between strength and accuracy-based fitness has been sought. In order to compare XCS with more traditional classifier systems, SB-XCS, XCS's strength-based twin, is introduced. SB-XCS is designed to

Contents

1	Introduction	1
1.1	Two Example Machine Learning Tasks	2
1.2	Types of Task	3
1.2.1	Supervised and Reinforcement Learning	3
1.2.2	Sequential and Non-sequential Decision Tasks	4
1.3	Two Challenges for Classifier Systems	4
1.3.1	Problem 1: Learning a Policy from Reinforcement	5
1.3.2	Problem 2: Generalisation	5
1.4	Solution Methods	6
1.4.1	Method 1: Reinforcement Learning Algorithms	6
1.4.2	Method 2: Evolutionary Algorithms	6
1.5	Learning Classifier Systems	6
1.5.1	The Tripartite LCS Structure	7
1.5.2	LCS = Policy Learning + Generalisation	7
1.5.3	Credit Assignment in Classifier Systems	8
1.5.4	Strength and Accuracy-based Classifier Systems	8
1.6	About the Book	9
1.6.1	Why Compare Strength and Accuracy?	10
1.6.2	Are LCS EC- or RL-based?	11
1.6.3	Moving in Design Space	14
1.7	Structure of the Book	16
2	Learning Classifier Systems	19
2.1	Types of Classifier Systems	21
2.1.1	Michigan and Pittsburgh LCS	21
2.1.2	XCS and Traditional LCS?	21
2.2	Representing Rules	22
2.2.1	The Standard Ternary Language	22
2.2.2	Other Representations	24
2.2.3	Summary of Rule Representation	25
2.2.4	Notation for Rules	25

2.3	XCS	25
2.3.1	Wilson's Motivation for XCS	26
2.3.2	Overview of XCS	27
2.3.3	Wilson's Explore/Exploit Framework	30
2.3.4	The Performance System	32
2.3.4.1	The XCS Performance System Algorithm	32
2.3.4.2	The Match Set and Prediction Array	32
2.3.4.3	Action Selection	34
2.3.4.4	Experience-weighting of System Prediction	34
2.3.5	The Credit Assignment System	35
2.3.5.1	The MAM Technique	35
2.3.5.2	The Credit Assignment Algorithm	36
2.3.5.3	Sequential and Non-sequential Updates	36
2.3.5.4	Parameter Update Order	37
2.3.5.5	XCS Parameter Updates	38
2.3.6	The Rule Discovery System	41
2.3.6.1	Random Initial Populations	41
2.3.6.2	Covering	41
2.3.6.3	The Niche Genetic Algorithm	43
2.3.6.4	Alternative Mutation Schemes	44
2.3.6.5	Triggering the Niche GA	45
2.3.6.6	Deletion of Rules	45
2.3.6.7	Classifier Parameter Initialisation	46
2.3.6.8	Subsumption Deletion	47
2.4	SB-XCS	47
2.4.1	Specification of SB-XCS	48
2.4.2	Comparison of SB-XCS and Other Strength LCS	51
2.5	Initial Tests of XCS and SB-XCS	52
2.5.1	The 6 Multiplexer	52
2.5.2	Woods2	55
2.6	Summary	60
3	How Strength and Accuracy Differ	63
3.1	Thinking about Complex Systems	63
3.2	Holland's Rationale for CS-1 and his Later LCS	65
3.2.1	Schema Theory	65
3.2.2	The Bucket Brigade	65
3.2.3	Schema Theory + Bucket Brigade = Adaptation	66
3.3	Wilson's Rationale for XCS	66
3.3.1	A Bias towards Accurate Rules	67
3.3.2	A Bias towards General Rules	67
3.3.3	Complete Maps	70
3.3.4	Summary	71
3.4	A Rationale for SB-XCS	71
3.5	Analysis of Populations Evolved by XCS and SB-XCS	71

3.5.1	SB-XCS	72
3.5.2	XCS	73
3.5.3	Learning Rate	76
3.6	Different Goals, Different Representations	76
3.6.1	Default Hierarchies	77
3.6.2	Partial and Best Action Maps	77
3.6.3	Complete Maps	79
3.6.4	What do XCS and SB-XCS Really Learn?	79
3.7	Complete and Partial Maps Compared	81
3.7.1	Advantages of Partial Maps	82
3.7.2	Disadvantages of Partial Maps	85
3.7.3	Complete Maps and Strength	91
3.7.4	Contrasting Complete and Partial Maps in RL Terminology	92
3.7.5	Summary of Comparison	92
3.8	Ability to Express Generalisations	93
3.8.1	Mapping Policies and Mapping Value Functions	93
3.8.2	Adapting the Accuracy Criterion	94
3.8.3	XCS-hard and SB-XCS-easy Functions	95
3.8.4	Summary of Generalisation and Efficiency	95
3.9	Summary	96
4	What Should a Classifier System Learn?	97
4.1	Representing Boolean Functions	99
4.1.1	Truth Tables	99
4.1.2	On-sets and Off-sets	99
4.1.3	Sigma Notation	100
4.1.4	Disjunctive Normal Form	100
4.1.5	Representing Functions with Sets of Rules	100
4.2	How Should a Classifier System Represent a Solution?	101
4.2.1	The Value of a Single Rule	102
4.2.2	The Value of a Set of Rules	103
4.2.3	Complete and Correct Representations	103
4.2.4	Minimal Representations	105
4.2.5	Non-overlapping Representations	107
4.2.6	Why XCS Prefers Non-overlapping Populations	108
4.2.7	Should we Prefer Non-overlapping Populations?	109
4.2.8	Optimal Rule Sets: [O]s	110
4.2.9	Conflicting Rules	111
4.2.10	Representation in XCS	111
4.3	How Should We Measure Performance?	112
4.3.1	Measures of Performance	112
4.3.2	Measures of Population State	113
4.3.3	Measuring Performance and Measuring State	114
4.3.4	New Population State Metrics	117

4.3.5	Testing XCS with %[PI]	118
4.3.6	Testing XCS with %[m-DNF]	120
4.3.7	Summary of Metrics and Properties	121
4.4	Summary	122
5	Prospects for Adaptation	125
5.1	Known Problems with Strength LCS	127
5.2	Methodology for Rule Type Analysis	128
5.3	Analysis of Rule Types	130
5.3.1	Correct and Incorrect Actions	130
5.3.2	Overgeneral Rules	131
5.3.3	Strong Overgeneral Rules	136
5.3.4	Fit Overgeneral Rules	137
5.3.5	Parallel Definitions of Strength and Fitness	138
5.4	When are Strong and Fit Overgenerals Possible?	139
5.4.1	Biases in the Reward Function are Relevant	140
5.4.2	Competition for Action Selection	140
5.4.3	Competition for Reproduction	142
5.5	Strong Overgenerals in XCS	142
5.5.1	Biases between Actions do not Produce Strong Overgenerals	144
5.5.2	Some Properties of Accuracy-based Fitness	144
5.6	Strong Overgenerals in SB-XCS	146
5.6.1	When are Strong Overgenerals Impossible in SB-XCS?	148
5.6.2	What Makes Strong Overgenerals Possible in SB-XCS?	148
5.7	Fit Overgenerals and the Survival of Rules under the GA	150
5.7.1	Comparison on an Unbiased Reward Function	150
5.7.2	Comparison on a Biased Reward Function	150
5.7.3	Discussion	151
5.8	Designing Strong and Fit Overgenerals for XCS	152
5.8.1	Biased Variance Functions	153
5.8.2	Empirical Results	153
5.8.3	Avoiding Fit Overgenerals	155
5.8.4	SB-XCS and Biased Variance Functions	156
5.9	Strong and Fit Undergeneral Rules	156
5.10	Why Bias the Reward Function?	157
5.10.1	Some State-actions are more Important than Others	158
5.10.2	A Rule Allocation Bias can Focus Resources	158
5.11	Rule Allocation Reconsidered	159
5.11.1	Knowing What Not to Do	159
5.11.2	Managing Exploration	160
5.11.3	Complete and Partial Maps Revisited	161
5.11.4	Alternatives to Biasing the Reward Function	161
5.11.5	Can SB-XCS Avoid Strong and Fit Overgenerals?	162
5.12	Sequential Tasks	162

5.12.1	The Need to Pass Values Back	163
5.12.2	The Need for Discounting	164
5.12.3	How Q-functions become Biased	165
5.12.4	Examples	165
5.12.5	Woods2 Revisited	166
5.12.6	When Will the Value Function be Unbiased?	173
5.13	What Tasks can we Solve with SB-XCS?	174
5.14	Extensions	175
5.14.1	Fitness Sharing	175
5.14.2	Other Factors Contributing to Strong Overgenerals ...	176
5.14.3	Qualitative and Quantitative Approaches	177
5.15	Summary	178
6	Classifier Systems and Q-learning	179
6.1	Classifier Systems and Q-learning	180
6.1.1	Q-learning in Classifier Systems	180
6.1.2	Is it Really Q-learning?	181
6.1.3	XCS is a Proper Generalisation of Tabular Q-learning ..	182
6.1.4	Summary	182
6.2	The GA-view and RL-view Revisited	183
6.2.1	How SB-XCS Determines Policies	183
6.2.2	How XCS Determines Policies	184
6.2.3	Three Approaches to Determining a Policy	185
6.2.4	The GA-view and the RL-view	185
6.2.5	Combining Evolution and Q-learning	186
6.3	XCS is Closer to Tabular Q-learning than to SB-XCS	188
6.4	Summary	189
7	Conclusion	191
7.1	The Capacities of Various Types of LCS	191
7.2	Contributions	192
7.3	The Take-home Message	195
7.4	Open Problems and Future Work	197
7.4.1	Fitness Sharing and Strength-based Fitness	197
7.4.2	Further Study of Accuracy-based Fitness	197
7.5	Concluding Remarks	198
7.5.1	The Moral of the Story: The Need for a Complex Systems Design Methodology	198
7.5.2	Classifier Systems and Reinforcement Learning	199
7.5.3	The Future	200

APPENDICES

A	Evaluation of Macroclassifiers	201
----------	---------------------------------------------	------------

B	Example XCS Cycle	203
B.1	The Performance System Algorithm	204
B.2	The Credit Assignment Algorithm	206
B.3	The Rule Discovery Algorithm	209
C	Learning from Reinforcement	213
C.1	Three Learning Paradigms	214
C.1.1	Supervised Learning	214
C.1.2	Reinforcement Learning	215
C.1.3	Unsupervised Learning	216
C.2	The Explore/Exploit Dilemma: a Feature of RL	216
C.3	Sequential and Non-sequential Tasks	218
C.3.1	Immediate Reward and Long-term Value	219
C.3.2	Sequential Decisions Imply RL	219
C.3.3	Episodic and Continuing Tasks	220
C.4	The Agent's Goal: Maximising Return	220
C.4.1	Return and Reward	220
C.4.2	Sequential Formulations of Return	221
C.5	Formalising RL Tasks	222
C.5.1	Environment	222
C.5.2	Learning Agent	223
C.5.3	Agent–environment Interaction	224
C.6	Summary	225
D	Generalisation Problems	227
D.1	Why Generalise?	228
D.1.1	The Curse of Dimensionality	228
D.1.2	The Need for Generalisation	228
D.2	Generalisation in RL	229
D.2.1	Generalising Over Policies and Value Functions	229
D.3	State Aggregation	230
D.4	State Space and Generalisation Space	230
D.5	Summary	230
E	Value Estimation Algorithms	233
E.1	The Value of State-actions	234
E.2	Non-sequential RL: Estimating Reward Functions	235
E.2.1	The Value of State-actions in Non-sequential Tasks	235
E.3	Estimating Expectations with Sample Averages	235
E.3.1	Incremental Updates	236
E.3.2	A General Form of Incremental Update	237
E.3.3	Setting <i>StepSize</i> in Incremental Updates	237
E.3.4	A Prediction Algorithm for Non-sequential RL	238
E.4	Sequential RL: Estimating Long-term Value Functions	238
E.4.1	Long-term Value Functions	239

E.4.2	The Value of State-actions in Sequential Tasks	241
E.4.3	The Value of a Policy	241
E.4.4	Estimating Values with Monte Carlo Methods	242
E.4.5	Estimating Values with Temporal Difference Methods	243
E.4.6	Russell and Norvig's Maze: A Sequential RL Task	245
E.4.7	Summary of Sequential RL	246
E.5	State Aggregation	246
E.5.1	Fixed and Adaptive Aggregation Schemes	246
E.5.2	The Value of Aggregations I: Return	247
E.5.3	The Value of Aggregations II: Predictive Utility	248
E.6	Storing Value Estimates	249
E.6.1	Storing Estimates of Aggregations	249
E.6.2	Sparse Estimators, Models and Search	250
E.6.3	Function Approximators	250
E.7	Summary	250
F	Generalised Policy Iteration Algorithms	251
F.1	Policy Improvement	252
F.2	Optimal Policies	253
F.3	Generalised Policy Iteration	253
F.3.1	How Well must we Evaluate a Policy?	254
F.3.2	Convergence Properties of GPI Control Algorithms	255
F.3.3	Initialising Value Functions	255
F.3.4	What Characterises GPI Algorithms?	255
F.4	State-value Functions	255
F.5	Summary	256
G	Evolutionary Algorithms	257
G.1	Evolution	258
G.2	Elements of EAs	260
G.2.1	A Generic EA	260
G.2.2	Population-based Search	261
G.2.3	Fitness Functions	261
G.2.4	Probabilistic Selection of Parents	261
G.2.5	Genetic Operators	262
G.2.6	Replacement	263
G.3	EAs as Search	263
G.3.1	Local and Global Optima	264
G.4	The Generalisation Problem	264
G.5	Niching and Mating Restriction	265
G.5.1	Fitness Sharing	266
G.5.2	Crowding	267
G.5.3	Mating Restriction	267
G.6	RL with EAs	267
G.6.1	Non-associative RL with an EA	268

G.6.2	Associative RL with an EA	269
G.6.3	Sequential RL with an EA	271
G.7	Comparing GPI and EA Methods for RL	272
G.7.1	Similarities between GPI and EA Methods	272
G.8	Summary	273
H	The Origins of Sarsa	275
H.1	Modified Connectionist Q-learning	276
H.2	ZCS's Implicit Bucket Brigade	276
H.3	Who Invented Sarsa?	277
I	Notation	279
	References	283
	Index	303



<http://www.springer.com/978-1-85233-770-4>

Strength or Accuracy: Credit Assignment in Learning
Classifier Systems

Kovacs, T.

2004, XVI, 307 p., Hardcover

ISBN: 978-1-85233-770-4