Contents

1 Introduction 1
1.1 Two Example Machine Learning Tasks 2
1.2 Types of Task 4
1.2.1 Supervised and Reinforcement Learning 4
1.2.2 Sequential and Non-Sequential Decision Tasks 4
1.3 Two Challenges for Classifier Systems 5
1.3.1 Problem 1: Learning a Policy from Reinforcement . . . 6
1.3.2 Problem 2: Generalisation 6
1.4 Solution Methods 6
1.4.1 Method 1: Reinforcement Learning Algorithms 6
1.4.2 Method 2: Evolutionary Algorithms 7
1.5 Learning Classifier Systems 7
1.5.1 The Tripartite LCS Structure 7
1.5.2 LCS = Policy Learning + Generalisation 8
1.5.3 Credit Assignment in Classifier Systems 9
1.5.4 Strength and Accuracy-based Classifier Systems 10
1.6 About the Thesis 10
1.6.1 Why Compare Strength and Accuracy? 11
1.6.2 Are LCS EC- or RL-based? 12
1.6.3 Moving in Design Space 15
1.7 Structure of the Thesis 17

2 Learning Classifier Systems 21
2.1 Types of Classifier Systems 23
2.1.1 Michigan and Pittsburgh LCS 23
2.1.2 XCS and Traditional LCS? 24
2.2 Representing Rules 25
2.2.1 The Standard Ternary Language 25
2.2.2 Other Representations 26
2.2.3 Summary of Rule Representation 28
2.2.4 Notation for Rules 28

2.3 XCS 28
2.3.1 Wilson's Motivation for XCS 29
2.3.2 Overview of XCS 30
2.3.3 Wilson's Explore/Exploit Framework 33
2.3.4 The Performance System 35
2.3.5 The Credit Assignment System 38
2.3.6 The Rule Discovery System 45
2.4 SB–XCS 52
2.4.1 Specification of SB–XCS 53
2.4.2 Comparison of SB–XCS and Other Strength LCS . . . 56
2.5 Initial Tests of XCS and SB–XCS 57
2.5.1 The 6 Multiplexer 57
2.5.2 Woods2 61
2.6 Summary 66

3 How Strength and Accuracy Differ 69

3.1 Thinking about Complex Systems 70
3.2 Holland's Rationale for CS-1 and his Later LCS 71
3.2.1 Schema Theory 71
3.2.2 The Bucket Brigade 72
3.2.3 Schema Theory + Bucket Brigade = Adaptation 73
3.3 Wilson's Rationale for XCS 73
3.3.1 A Bias towards Accurate Rules 73
3.3.2 A Bias towards General Rules 74
3.3.3 Complete Maps 77
3.3.4 Summary 78
3.4 A Rationale for SB–XCS 78
3.5 Analysis of Populations Evolved by XCS and SB–XCS 78
3.5.1 SB–XCS 79
3.5.2 XCS 81
3.6 Different Goals, Different Representations 83
3.6.1 Default Hierarchies 84
3.6.2 Partial and Best Action Maps 85
3.6.3 Complete Maps 86
3.6.4 What do XCS and SB–XCS Really Learn? 87
3.7 Complete and Partial Maps Compared 89
3.7.1 Advantages of Partial Maps 89
3.7.2 Disadvantages of Partial Maps 92
3.7.3 Complete Maps and Strength 99
3.7.4 Contrasting Complete and Partial Maps in RL Terminology 100

3.7.5 Summary of Comparison 100
3.8 Ability to Express Generalisations 100
3.8.1 Mapping Policies and Mapping Value Functions 101
3.8.2 Adapting the Accuracy Criterion 102
3.8.3 XCS-hard and SB–XCS-easy Functions 103
3.8.4 Summary of Generalisation and Efficiency 103
3.9 Summary 104

4 What Should a Classifier System Learn? 105
4.1 Representing Boolean Functions 107
4.1.1 Truth Tables 107
4.1.2 On-sets and Off-sets 108
4.1.3 Sigma Notation 108
4.1.4 Disjunctive Normal Form 108
4.1.5 Representing Functions with Sets of Rules 109
4.2 How Should a Classifier System Represent a Solution? 110
4.2.1 The Value of a Single Rule 111
4.2.2 The Value of a Set of Rules 111
4.2.3 Complete and Correct Representations 112
4.2.4 Minimal Representations 114
4.2.5 Non-overlapping Representations 117
4.2.6 Optimal Rule Sets: [O]s 120
4.2.7 Conflicting Rules 120
4.2.8 Representation in XCS 120
4.3 How Should We Measure Performance? 121
4.3.1 Measures of Performance 122
4.3.2 Measures of Population State 123
4.3.3 Measuring Performance and Measuring State 124
4.3.4 New Population State Metrics 127
4.3.5 Testing XCS with %[PI] 128
4.3.6 Testing XCS with %[m-DNF] 130
4.3.7 Summary of Metrics and Properties 131
4.4 Summary 132

5 Prospects for Adaptation 135
5.1 Known Problems with Strength LCS 137
5.2 Methodology for Rule Type Analysis 139
5.3 Analysis of Rule Types 141
5.3.1 Correct and Incorrect Actions 141
5.3.2 Overgeneral Rules 142
5.3.3 Strong Overgeneral Rules 148

5.3.4 Fit Overgeneral Rules 149
5.3.5 Parallel Definitions of Strength and Fitness 149
5.4 When are Strong and Fit Overgenerals Possible? 150
5.4.1 The Reward Function is Relevant 151
5.4.2 Competition for Action Selection 153
5.4.3 Competition for Reproduction 154
5.5 Strong Overgenerals in XCS 155
5.5.1 Biases between Actions do not Produce Strong Overgenerals 156
5.5.2 Some Properties of Accuracy-Based Fitness 157
5.6 Strong Overgenerals in SB–XCS 159
5.6.1 When are Strong Overgenerals Impossible in SB–XCS? 160
5.6.2 What Makes Strong Overgenerals Possible in SB–XCS? 161
5.7 Fit Overgenerals and the Survival of Rules under the GA . . . 163
5.7.1 Comparison on an Unbiased Reward Function 164
5.7.2 Comparison on a Biased Reward Function 164
5.7.3 Discussion 164
5.8 Designing Strong and Fit Overgenerals for XCS 166
5.8.1 Biased Variance Functions 166
5.8.2 Empirical Results 168
5.8.3 Avoiding Fit Overgenerals 168
5.8.4 SB–XCS and Biased Variance Functions 170
5.9 Strong and Fit Undergeneral Rules 170
5.10 Why Bias the Reward Function? 171
5.10.1 Some State-actions are more Important than Others . . 172
5.10.2 A Rule Allocation Bias can Focus Resources 172
5.11 Rule Allocation Reconsidered 173
5.11.1 Knowing What Not to Do 174
5.11.2 Managing Exploration 174
5.11.3 Complete and Partial Maps Revisited 175
5.11.4 Alternatives to Biasing the Reward Function 175
5.11.5 Can SB–XCS Avoid Strong and Fit Overgenerals? . . . 176
5.12 Sequential Tasks 176
5.12.1 The Need to Pass Values Back 178
5.12.2 The Need for Discounting 179
5.12.3 How Q-functions become Biased 179
5.12.4 Examples 180
5.12.5 Woods2 Revisited 181
5.12.6 When Will the Value Function be Unbiased? 188
5.13 What Tasks can we Solve with SB–XCS? 190
5.14 Extensions 191

5.14.1 Fitness Sharing 191
5.14.2 Other Factors Contributing to Strong Overgenerals . . 192
5.14.3 Qualitative and Quantitative Approaches 193
5.15 Summary 194

6 Classifier Systems and Q-Learning 197
6.1 Classifier Systems and Q-Learning 198
6.1.1 Q-Learning in Classifier Systems 198
6.1.2 Is it Really Q-Learning? 199
6.1.3 XCS is a Proper Generalisation of Tabular Q-Learning 200
6.1.4 Summary 201
6.2 The GA-View and RL-View Revisited 201
6.2.1 How SB–XCS Determines Policies 202
6.2.2 How XCS Determines Policies 203
6.2.3 Three Approaches to Determining a Policy 204
6.2.4 The GA-View and the RL-View 204
6.2.5 Combining Evolution and Q-learning 205
6.3 XCS is Closer to Tabular Q-Learning than to SB–XCS 208
6.4 Summary 208

7 Conclusion 209
7.1 The Capacities of Various Types of LCS 209
7.2 Contributions 210
7.3 The Take-home Message 214
7.4 Open Problems and Future Work 216
7.4.1 Fitness Sharing and Strength-based Fitness 216
7.4.2 Further Study of Accuracy-based Fitness 217
7.5 Concluding Remarks 217
7.5.1 The Moral of the Story: The Need for a Complex Systems Design Methodology 218
7.5.2 Classifier Systems and Reinforcement Learning 219
7.5.3 The Future 219

APPENDICES 220

A Evaluation of Macroclassifiers 221

B Example XCS Cycle 223
B.1 The Performance System Algorithm 224
B.2 The Credit Assignment Algorithm 226

B.3 The Rule Discovery Algorithm 229

C **Learning from Reinforcement** **233**

C.1 Three Learning Paradigms 234

C.1.1 Supervised Learning 234

C.1.2 Reinforcement Learning 235

C.1.3 Unsupervised Learning 237

C.2 The Explore/Exploit Dilemma: a Feature of RL 237

C.3 Sequential and Non-sequential Tasks 238

C.3.1 Immediate Reward and Long-term Value 239

C.3.2 Sequential Decisions Imply RL 240

C.3.3 Episodic and Continuing Tasks 240

C.4 The Agent's Goal: Maximising Return 241

C.4.1 Return and Reward 241

C.4.2 Sequential Formulations of Return 242

C.5 Formalising RL Tasks 243

C.5.1 Environment 243

C.5.2 Learning Agent 244

C.5.3 Agent-environment Interaction 245

C.6 Summary 246

D **Generalisation Problems** **247**

D.1 Why Generalise? 248

D.1.1 The Curse of Dimensionality 248

D.1.2 The Need for Generalisation 249

D.2 Generalisation in RL 249

D.2.1 Generalising Over Policies and Value Functions 250

D.3 State Aggregation 250

D.4 State Space and Generalisation Space 250

D.5 Summary 251

E **Value Estimation Algorithms** **253**

E.1 The Value of State-actions 254

E.2 Non-sequential RL: Estimating Reward Functions 255

E.2.1 The Value of State-actions in Non-sequential Tasks . . 255

E.3 Estimating Expectations with Sample Averages 256

E.3.1 Incremental Updates 256

E.3.2 A General Form of Incremental Update 257

E.3.3 Setting *StepSize* in Incremental Updates 258

E.3.4 A Prediction Algorithm for Non-sequential RL 258

E.4 Sequential RL: Estimating Long-term Value Functions 259

E.4.1 Long-term Value Functions 259
E.4.2 The Value of State-actions in Sequential Tasks 261
E.4.3 The Value of a Policy 262
E.4.4 Estimating Values with Monte Carlo Methods 262
E.4.5 Estimating Values with Temporal Difference Methods . 263
E.4.6 Russell and Norvig's Maze: A Sequential RL Task . . . 266
E.4.7 Summary of Sequential RL 267
E.5 State Aggregation 267
E.5.1 Fixed and Adaptive Aggregation Schemes 267
E.5.2 The Value of Aggregations I: Return 269
E.5.3 The Value of Aggregations II: Predictive Utility 269
E.6 Storing Value Estimates 270
E.6.1 Storing Estimates of Aggregations 270
E.6.2 Sparse Estimators, Models and Search 271
E.6.3 Function Approximators 271
E.7 Summary 272

F Generalised Policy Iteration Algorithms 273
F.1 Policy Improvement 274
F.2 Optimal Policies 275
F.3 Generalised Policy Iteration 275
F.3.1 How Well must we Evaluate a Policy? 276
F.3.2 Convergence Properties of GPI Control Algorithms . . 277
F.3.3 Initialising Value Functions 277
F.3.4 What Characterises GPI Algorithms? 277
F.4 State-value Functions 278
F.5 Summary 278

G Evolutionary Algorithms 281
G.1 Evolution 282
G.2 Elements of EAs 284
G.2.1 A Generic EA 284
G.2.2 Population-based Search 285
G.2.3 Fitness Functions 285
G.2.4 Probabilistic Selection of Parents 286
G.2.5 Genetic Operators 287
G.2.6 Replacement 288
G.3 EAs as Search 288
G.3.1 Local and Global Optima 289
G.4 The Generalisation Problem 290
G.5 Niching and Mating Restriction 290

G.5.1 Fitness Sharing 290
G.5.2 Crowding 291
G.5.3 Mating Restriction 291
G.6 RL with EAs 292
G.6.1 Non-associative RL with an EA 292
G.6.2 Associative RL with an EA 294
G.6.3 Sequential RL with an EA 296
G.7 Comparing GPI and EA Methods for RL 296
G.7.1 Similarities between GPI and EA Methods 297
G.8 Summary 298

H The Origins of Sarsa 299
H.1 Modified Connectionist Q-learning 300
H.2 ZCS's Implicit Bucket Brigade 300
H.3 Who Invented Sarsa? 301

I Notation 303

References 307

Index 336

Errata 343