Preface

This textbook provides an accessible introduction to Learning Classifier Systems (LCSs) for undergraduate/postgraduate students, data analysts, and machine learning practitioners alike. We aim to tackle the following questions through the lens of simple example problems: (1) How do LCSs work and how can they be implemented? (2) What types of problems have or can they be applied to? (3) What makes LCS algorithms unique and advantageous compared to other machine learners? (4) What challenges or disadvantages exist? (5) What variations of LCS algorithms exist? (6) What resources exist to support the development of new LCS algorithms, or the application of existing ones?

The term LCS has been used to describe a family of machine learning (ML) algorithms that emerged from a founding concept designed to model complex adaptive systems (e.g. the economy, weather, or the human brain). The LCS concept has enjoyed over 40 years of active research from a small but dedicated community and LCS algorithms have repeatedly demonstrated their unique value across a diverse and growing set of applications. Despite this history, LCS algorithms are still largely unknown and underutilised among ML approaches. This is true for both types of problems to which LCS algorithms are most commonly applied (i.e. reinforcement learning and supervised learning).

Reinforcement learning problems only provide the learner with occasional feedback in the form of reward/punishment. This type of learning has close ties to the broader field of artificial intelligence and has been applied to tasks such as behavior modeling, maze navigation, and game play. On the other hand, *supervised learning* problems provide the learner with the correct decision as input. This type of learning is commonly applied to data science tasks such as predictive modeling (i.e. classification or regression).

The name *Learning Classifier System* is a bit odd/misleading since there are many ML algorithms that learn to classify (such as decision trees or support vector machines) but are not LCSs. A slightly more general term that better represents LCS algorithms is *Rule-Based Machine Learning* (RBML). This term encompasses the two main families of LCS algorithms (i.e. *Michigan-style* and *Pittsburgh-style*), as well as Association Rule Learning and Artificial Immune Systems, which are not LCS algorithms. The LCS concept was developed by John Holland in the 1970s around the same time that he popularised what is now known as a *Genetic Algorithm* (GA). LCSs typically incorporate a GA, and as such are sometimes even more generally referred to as *Genetics-Based Machine Learning* (GBML). Depending on context, these three terms (LCS, RBML, and GBML) can be used interchangeably.

At a high level, LCS algorithms all combine a *discovery component* (typically driven by *Evolutionary Computation* (EC) methods, such as a genetic algorithm) and a *learning component* that tracks accuracy and/or handles credit assignment in order to improve performance through the acquisition of experience. A basic understanding of EC would be a useful prerequisite for this book. In short, EC is a field that studies algorithms inspired by the principles of Darwinian evolution.

In our opinion, the most distinguishing feature of LCSs, and RBML in general, is that the 'model' output by the system is a set of rules that each 'cover' (i.e. are relevant to) only a subset of the possible inputs to the problem at hand. Each rule represents an *IF:THEN* expression that links specific state conditions with an action/class. For example, *IF 'red' AND 'octagon' THEN 'stop-sign'* might be a rule learned for classifying types of traffic signs. Notice that this rule does not constitute a complete model, but rather is only part of the collaborative ruleset required to accurately and generally classify the variety of road signs based on available features such as color, shape, or size. This property allows LCS algorithms to challenge a nearly ubiquitous paradigm of machine learning, i.e. that a single 'best' model will be found. LCS algorithms instead learn a 'distributed solution' or a 'map' of the problem space represented by a ruleset that implicitly breaks complex problems into simpler pieces. This property is the main reason why LCS algorithms can (1) flexibly handle very different problem domains, (2) adapt to new input, (3) model complex patterns such as non-linear feature interactions (i.e. epistasis) and heterogeneous associations, and (4) be applied to either single-step or multi-step problems.

Additionally, LCS algorithms have the following advantages: (1) They are fundamentally model free, i.e. they make few or no assumptions about the underlying problem domain, (2) their rules are intuitively human interpretable, unlike so-called 'black box' ML algorithms such as artificial neural networks or random forests, (3) they produce solutions that are implicitly multi-objective, with evolutionary pressures driving maximally accurate and general rules, and (4) they resemble ensemble learners, which tend to make more accurate and reliable predictions, particularly when prior problem knowledge is unavailable. This book will highlight the types of problems to which LCS algorithms have been shown to be particularly well suited, e.g. those with epistasis and heterogeneity.

Theoretical understanding of the LCS approach has improved, but an accepted theory does not yet exist. This is probably due to the interactive complexity and underlying stochastic nature of LCSs. Whether it is even possible to include convergence proofs is debatable, although such a proof would be beneficial in cross-disciplinary acceptance and adoption.

This book is intended as a jumping-off point, and does not include a detailed history of LCSs, nor does it explore many of the cutting-edge advancements available in the field today. Many great researchers, papers, and ideas will not be cited. Instead, it addresses an outstanding need for a simple introduction to the LCS concept, which can seem a bit tricky to grasp compared to other ML algorithms. This is due to the unusual learning paradigm offered by LCSs, as well as the multiple interacting components that make up these algorithms. Conveniently, the components of an LCS can be exchanged, added, or removed (like algorithmic Lego building blocks), yielding a framework with the problem versatility of a Swiss Army knife. To facilitate comprehension of how LCSs operate, and how they can be implemented, we have paired this book with an educational version of LCS, named *eLCS*, coded simply in Python. Grant support from the National Institutes of Health (R01 AI116794), and the Victoria University of Wellington (204021) helped make this book possible. Please enjoy!

Ryan Urbanowicz, University of Pennsylvania, USA
Will Browne, Victoria University of Wellington, NZ

http://www.springer.com/978-3-662-55006-9

Introduction to Learning Classifier Systems
Urbanowicz, R.J.; Browne, W.N.
2017, XIII, 123 p. 27 illus., 4 illus. in color., Softcover
ISBN: 978-3-662-55006-9