SpringerBriefs in Intelligent Systems

Artificial Intelligence, Multiagent Systems, and Cognitive Robotics

This series covers the entire research and application spectrum of intelligent systems, including artificial intelligence, multiagent systems, and cognitive robotics. Typical texts for publication in the series include, but are not limited to, state-of-the-art reviews, tutorials, summaries, introductions, surveys, and in-depth case and application studies of established or emerging fields and topics in the realm of computational intelligent systems. Essays exploring philosophical and societal issues raised by intelligent systems are also very welcome.

More information about this series at http://www.springer.com/series/11845

Ryan J. Urbanowicz · Will N. Browne

Introduction to Learning Classifier Systems

Springer

Ryan J. Urbanowicz
Department of Biostatistics, Epidemiology, and Informatics, Perelman School of Medicine
University of Pennsylvania
Philadelphia, PA
USA

Will N. Browne
School of Engineering and Computer Science
Victoria University of Wellington
Wellington
New Zealand

ISSN 2196-548X ISSN 2196-5498 (electronic)
SpringerBriefs in Intelligent Systems
ISBN 978-3-662-55006-9 ISBN 978-3-662-55007-6 (eBook)
DOI 10.1007/978-3-662-55007-6

Library of Congress Control Number: 2017950033

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag GmbH Germany
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

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

Contents

Acronyms and Glossary

Acronyms

Acronyms used by related methods and fields of study.

AI	Artificial Intelligence - intelligence exhibited by machines. A flexible rational agent that perceives its environment and takes actions to maximise its chance of success.
DM	Data Mining - A main component of knowledge discovery in databases, where patterns and knowledge contained in data are discovered.
EC	Evolutionary Computation - Global search method based on the principles of Darwinian selection.
EDA	Estimation of Distribution Algorithm - Builds a probabilistic model of the solution.
EML	Evolutionary Machine Learning - Machine learning using Darwinian principles: Modern name for the overarching field that includes LCS (replacing the term GBML).
GA	Genetic Algorithm - Both the name of the field of genetic algorithms and the method used to discover hypothesised better rules in LCS (see RD).
GBML	Genetics-Based Machine Learning - The term used to identify any ML approach with a genetics-based evolutionary component (superseded by EML).
LCS	Learning Classifier System - The original name for the concept of an artificial intelligence technique that builds models of patterns inherent in the data through global search of pattern structures (e.g. rules created through evolutionary computation) and local learning of pattern utility (e.g. pattern fitness through supervised or reinforcement learning). Note that 'LCSs' will be used throughout this book to refer to the plural of 'LCS'.
ML	Machine Learning - A subfield of computer science emerging from pattern recognition and artificial intelligence exploring algorithms that can learn from and make predictions about data.
RBML	Rule-Based Machine Learning - A term for machine learning algorithms that employ rules or classifiers in modeling. RBML includes all types of LCS, along with Association Rule Learning, and Artificial Immune Systems.
RD	Rule Discovery - Mechanisms that can introduce new rules in LCS.
RL	Reinforcement Learning - Learning from environmental reward payoff only.
SL	Supervised Learning - Learning from a teacher in the environment.

Key LCS Terms

There are a few terms that are tailored to the field of LCSs that are not in common use in other branches of AI.

action	The endpoint, consequent, output, or THEN portion of the 'if ... then ...' rule expression evolved by an LCS.
classifier	A rule plus supporting statistics learned from environmental interaction.
condition	The specified feature states, antecedent, input, or IF portion of the 'if ... then ...' rule expression evolved by an LCS.
coverage	The sample space matched by the condition portion of a rule.
niche	An area of the sample space in the domain where the neighboring instances share a common property, e.g. same class.
rule	An 'if <condition> then <action>' expression evolved by an LCS.
sample space	The unique instances (states) available from the problem domain.
search space	The unique rules that can be created; the search space is linked to the sample space together with the chosen alphabet/rule representation of the LCS.

Rulesets

Sets of rules in LCS are conventionally indicated by square brackets '[]' rather than curly brackets '{}'. The symbol for each set is italicised, indicating a variable, given that the content of these sets will vary during evolution. Note that while italics ought to be used, they are often omitted by convention in the LCS literature.

$[P]$	Population - the set of all classifiers in the LCS (utilised in both reinforcement and supervised learning).
$[M]$	Match Set - the set of all classifiers with conditions that match the environmental state. Matching rules can recommend different actions, thus $[M]$ is a superset of $[A]$ (in reinforcement learning), and $[M]$ is a superset of both $[C]$ and $[I]$ (in supervised learning).
$[A]$	Action Set - the set of all classifiers with matching conditions (i.e. that are in $[M]$) that also recommend the action that the system selected to affect the environment (used in reinforcement learning).
$[C]$	Correct Set - the set of all classifiers with matching conditions (i.e. that are in $[M]$) that also recommend the **correct** action obtained from the environment (used in supervised learning).
$[I]$	Incorrect Set - the set of all classifiers with matching conditions (i.e. that are in $[M]$) that also recommend an **incorrect** action obtained from the environment (used in supervised learning).

Variants of LCS Algorithms

Names of LCS variants (there have been many in the history of LCS, so this is only a subsample).

BioHEL	Bioinformatics-oriented Hierarchical Evolutionary Learning - BioHEL is designed to handle large-scale, e.g. bioinformatic, datasets using a meta-representation.
eLCS	Educational LCS - A bare-bones LCS specifically designed to complement this book and facilitate understanding of LCS implementation. Not intended to function optimally on real-world problems. (Users are welcome to develop eLCS further and advance the academic field under the Creative Commons license).
ExSTraCS	Extended Supervised Tracking and Classifying System - ExSTraCS extends the UCS framework adding expert-knowledge-guided learning, attribute tracking for heterogeneous subgroup identification, and a number of other heuristics to handle complex, noisy, and larger-scale (e.g. bioinformatic) data mining.
GAssist	Genetic clASSIfier sySTem - Accuracy-based Pittsburgh learning classifier system with default action, incremental update and novel representation. Now superseded by BioHEL.
SCS	Simple Classifier System - now outdated simple LCS in Goldberg's 1989 book.
UCS	sUpervised Classifier System - UCS is based on the XCS framework but specifically adapted to supervised learning. Unfortunately SCS had already been taken and SuCS might not have been so successful (note: Unsupervised Learning Classifier Systems are very rare in the LCS field).
XCS	XCS - Michigan-style, accuracy-based, niche-based LCS that has become the most widely used system in the field. Designed to handle both reinforcement learning and supervised learning problems. Back acronymed to be eXtended Classifier System.
ZCS	Zeroth-level Classifier System - Revolutionary strength-based classifier system with implicit bucket brigade update mechanism.