Studies in Computational Intelligence

Volume 673

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Wojciech Wieczorek

Grammatical Inference

Algorithms, Routines and Applications

Springer

Wojciech Wieczorek
Institute of Computer Science
University of Silesia Faculty of Computer
Science and Materials Science
Sosnowiec
Poland

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-46800-6 ISBN 978-3-319-46801-3 (eBook)
DOI 10.1007/978-3-319-46801-3

Library of Congress Control Number: 2016952872

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Grammatical inference, the main topic of this book, is a scientific area that lies at the intersection of multiple fields. Researchers from computational linguistics, pattern recognition, machine learning, computational biology, formal learning theory, and many others have their own contribution. Therefore, it is not surprising that the topic has also a few other names such as grammar learning, automata inference, grammar identification, or grammar induction. To simplify the location of present contribution, we can divide all books relevant to grammatical inference into three groups: theoretical, practical, and applicable. In greater part this book is practical, though one can also find the elements of learning theory, combinatorics on words, the theory of automata and formal languages, plus some reference to real-life problems.

The purpose of this book is to present old and modern methods of grammatical inference from the perspective of practitioners. To this end, the Python programming language has been chosen as the way of presenting all the methods. Included listings can be directly used by the paste-and-copy manner to other programs, thus students, academic researchers, and programmers should find this book as the valuable source of ready recipes and as an inspiration for their further development.

A few issues should be mentioned regarding this book: an inspiration to write it, a key for the selection of described methods, arguments for selecting Python as an implementation language, typographical notions, and where the reader can send any critical remarks about the content of the book (subject–matter, listings etc.).

There is a treasured book entitled "Numerical recipes in C", in which along with the description of selected numerical methods, listings in C language are provided. The reader can copy and paste the fragments of the electronic version of the book in order to produce executable programs. Such an approach is very useful. We can find an idea that lies behind a method and immediately put it into practice. It is a guiding principle that accompanied writing the present book.

For the selection of methods, we try to keep balance between importance and complexity. It means that we introduced concepts and algorithms which are essential to the GI practice and theory, but omitted that are too complicated or too long to present them as a ready-to-use code. Thanks to that, the longest program included in the book is no more than a few pages long.

As far as the implementation language is concerned, the following requirements had to be taken into account: simplicity, availability, the property of being firmly established, and allowing the use of wide range of libraries. Python and FSharp programming languages were good candidates. We decided to choose IronPython (an implementation of Python) mainly due to its integration with the optimization modeling language. We use a monospaced (fixed-pitch) font for the listings of programs, while the main text is written using a proportional font. In listings, Python keywords are in bold.

The following persons have helped the author in preparing the final version of this book by giving valuable advice. I would like to thank (in alphabetical order): Prof. Z.J. Czech (Silesian University of Technology), Dr. P. Juszczuk, Ph.D. student A. Nowakowski, Dr. R. Skinderowicz, and Ph.D. student L. Strak (University of Silesia).

Sosnowiec, Poland Wojciech Wieczorek
2016

Contents

Acronyms

CFG	Context-free grammar
CGT	Combinatorial game theory
CNF	Chomsky normal form
CNF	Conjunctive normal form
CSP	Constraint satisfaction problem
DFA	Deterministic finite automaton
DNF	Disjunctive normal form
EDSM	Evidence driven state merging
GA	Genetic algorithm
GI	Grammatical inference
GNF	Greibach normal form
ILP	Integer linear programming
LP	Linear programming
MDL	Minimum description length
MILP	Mixed integer linear programming
NFA	Non-deterministic finite automaton
NLP	Non-linear programming
NP	Non-deterministic polynomial time
OGF	Ordinary generating function
OML	Optimization modeling language
PTA	Prefix tree acceptor
RPNI	Regular positive and negative inference
SAT	Boolean satisfiability problem
TSP	Traveling salesman problem
XML	Extensible markup language