

SpringerBriefs in Computer Science

Series editors

Stan Zdonik, Brown University, Providence, USA
Shashi Shekhar, University of Minnesota, Minneapolis, USA
Jonathan Katz, University of Maryland, College Park, USA
Xindong Wu, University of Vermont, Burlington, USA
Lakhmi C. Jain, University of South Australia, Adelaide, Australia
David Padua, University of Illinois Urbana-Champaign, Urbana, USA
Xuemin (Sherman) Shen, University of Waterloo, Waterloo, Canada
Borko Furht, Florida Atlantic University, Boca Raton, USA
V.S. Subrahmanian, University of Maryland, College Park, USA
Martial Hebert, Carnegie Mellon University, Pittsburgh, USA
Katsushi Ikeuchi, University of Tokyo, Tokyo, Japan
Bruno Siciliano, Università di Napoli Federico II, Napoli, Italy
Sushil Jajodia, George Mason University, Fairfax, USA
Newton Lee, Tujunga, USA

More information about this series at <http://www.springer.com/series/10028>

Rodrigo C. Barros · André C.P.L.F. de Carvalho
Alex A. Freitas

Automatic Design of Decision-Tree Induction Algorithms

 Springer

Rodrigo C. Barros
Faculdade de Informática
Pontifícia Universidade Católica do Rio
Grande do Sul
Porto Alegre, RS
Brazil

Alex A. Freitas
School of Computing
University of Kent
Canterbury, Kent
UK

André C.P.L.F. de Carvalho
Instituto de Ciências Matemáticas e de
Computação
Universidade de São Paulo
São Carlos, SP
Brazil

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-14230-2 ISBN 978-3-319-14231-9 (eBook)
DOI 10.1007/978-3-319-14231-9

Library of Congress Control Number: 2014960035

Springer Cham Heidelberg New York Dordrecht London

© The Author(s) 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

*This book is dedicated to my family:
Alessandra, my wife;
Marta and Luís Fernando, my parents;
Roberta, my sister; Gael, my godson;
Lygia, my grandmother.*

Rodrigo C. Barros

*To Valeria, my wife, and to Beatriz,
Gabriela and Mariana, my daughters.*

André C.P.L.F. de Carvalho

To Jie, my wife.

Alex A. Freitas

Contents

1	Introduction	1
	1.1 Book Outline	4
	References.	5
2	Decision-Tree Induction	7
	2.1 Origins	7
	2.2 Basic Concepts	8
	2.3 Top-Down Induction	9
	2.3.1 Selecting Splits.	11
	2.3.2 Stopping Criteria	29
	2.3.3 Pruning	30
	2.3.4 Missing Values.	36
	2.4 Other Induction Strategies	37
	2.5 Chapter Remarks	40
	References.	40
3	Evolutionary Algorithms and Hyper-Heuristics	47
	3.1 Evolutionary Algorithms	47
	3.1.1 Individual Representation and Population Initialization.	49
	3.1.2 Fitness Function	51
	3.1.3 Selection Methods and Genetic Operators	52
	3.2 Hyper-Heuristics	54
	3.3 Chapter Remarks	56
	References.	56
4	HEAD-DT: Automatic Design of Decision-Tree Algorithms.	59
	4.1 Introduction	60
	4.2 Individual Representation.	61
	4.2.1 Split Genes	61
	4.2.2 Stopping Criteria Genes.	63

4.2.3	Missing Values Genes	63
4.2.4	Pruning Genes	64
4.2.5	Example of Algorithm Evolved by HEAD-DT	66
4.3	Evolution	67
4.4	Fitness Evaluation	69
4.5	Search Space	72
4.6	Related Work	73
4.7	Chapter Remarks	74
	References	75
5	HEAD-DT: Experimental Analysis	77
5.1	Evolving Algorithms Tailored to One Specific Data Set	78
5.2	Evolving Algorithms from Multiple Data Sets	83
5.2.1	The Homogeneous Approach	84
5.2.2	The Heterogeneous Approach	99
5.2.3	The Case of Meta-Overfitting	121
5.3	HEAD-DT's Time Complexity	123
5.4	Cost-Effectiveness of Automated Versus Manual Algorithm Design	123
5.5	Examples of Automatically-Designed Algorithms	125
5.6	Is the Genetic Search Worthwhile?	126
5.7	Chapter Remarks	127
	References	139
6	HEAD-DT: Fitness Function Analysis	141
6.1	Performance Measures	141
6.1.1	Accuracy	142
6.1.2	F-Measure	142
6.1.3	Area Under the ROC Curve	143
6.1.4	Relative Accuracy Improvement	143
6.1.5	Recall	144
6.2	Aggregation Schemes	144
6.3	Experimental Evaluation	145
6.3.1	Results for the Balanced Meta-Training Set	146
6.3.2	Results for the Imbalanced Meta-Training Set	156
6.3.3	Experiments with the Best-Performing Strategy	164
6.4	Chapter Remarks	169
	References	170
7	Conclusions	171
7.1	Limitations	172
7.2	Opportunities for Future Work	173
7.2.1	Extending HEAD-DT's Genome: New Induction Strategies, Oblique Splits, Regression Problems	173

7.2.2	Multi-objective Fitness Function	173
7.2.3	Automatic Selection of the Meta-Training Set	174
7.2.4	Parameter-Free Evolutionary Search	174
7.2.5	Solving the Meta-Overfitting Problem	175
7.2.6	Ensemble of Automatically-Designed Algorithms	175
7.2.7	Grammar-Based Genetic Programming	176
References.	176

Notations

T	A decision tree
\mathbf{X}	A set of instances
N_x	The number of instances in \mathbf{X} , i.e., $ \mathbf{X} $
\mathbf{x}^j	An instance— n -dimensional attribute vector $[x_1^j, x_2^j, \dots, x_n^j]$ —from \mathbf{X} , $j = 1, 2, \dots, N_x$
\mathbf{X}_t	A set of instances that reach node t
A	The set of n predictive (independent) attributes $\{a_1, a_2, \dots, a_n\}$
y	The target (class) attribute
Y	The set of k class labels $\{y_1, \dots, y_k\}$ (or k distinct values if y is continuous)
$y(x)$	Returns the class label (or target value) of instance $\mathbf{x} \in \mathbf{X}$
$a_i(x)$	Returns the value of attribute a_i from instance $\mathbf{x} \in \mathbf{X}$
$dom(a_i)$	The set of values attribute a_i can take
$ a_i $	The number of partitions resulting from splitting attribute a_i
$\mathbf{X}_{a_i=v_j}$	The set of instances in which attribute a_i takes a value contemplated by partition v_j . Edge v_j can refer to a nominal value, to a set of nominal values, or even to a numeric interval
$N_{v_j, \bullet}$	The number of instances in which attribute a_i takes a value contemplated by partition v_j , i.e., $ \mathbf{X}_{a_i=v_j} $
$\mathbf{X}_{y=y_l}$	The set of instances in which the class attribute takes the label (value) y_l
N_{\bullet, y_l}	The number of instances in which the class attribute takes the label (value) y_l , i.e., $ \mathbf{X}_{y=y_l} $
$N_{v_j \cap y_l}$	The number of instances in which attribute a_i takes a value contemplated by partition v_j and in which the target attribute takes the label (value) y_l
v_X	The target (class) vector $[N_{\bullet, y_1}, \dots, N_{\bullet, y_k}]$ associated to \mathbf{X}
p_y	The target (class) probability vector $[p_{\bullet, y_1}, \dots, p_{\bullet, y_k}]$
p_{\bullet, y_l}	The estimated probability of a given instance belonging to class y_l , i.e., $\frac{N_{\bullet, y_l}}{N_x}$

$p_{v_j, \bullet}$	The estimated probability of a given instance being contemplated by partition v_j , i.e., $\frac{N_{v_j, \bullet}}{N_x}$
$p_{v_j \cap y_l}$	The estimated joint probability of a given instance being contemplated by partition v_j and also belonging to class y_l , i.e., $\frac{N_{v_j \cap y_l}}{N_x}$
$p_{y_l v_j}$	The conditional probability of a given instance belonging to class y_l given that it is contemplated by partition v_j , i.e., $\frac{N_{v_j \cap y_l}}{N_{v_j, \bullet}}$
$p_{v_j y_l}$	The conditional probability of a given instance being contemplated by partition v_j given that it belongs to class y_l , i.e., $\frac{N_{v_j \cap y_l}}{N_{\bullet, y_l}}$
ζ_T	The set of nonterminal nodes in decision tree T
λ_T	The set of terminal nodes in decision tree T
\aleph_T	The set of nodes in decision tree T , i.e., $\aleph_T = \zeta_T \cup \lambda_T$
$T^{(t)}$	A (sub)tree rooted in node t
$E^{(t)}$	The number of instances in t that do not belong to the majority class of that node