

Coding Ockham's Razor

Lloyd Allison

Coding Ockham's Razor



Springer

Lloyd Allison
Faculty of Information Technology
Monash University
Melbourne, Victoria, Australia

ISBN 978-3-319-76432-0 ISBN 978-3-319-76433-7 (eBook)
<https://doi.org/10.1007/978-3-319-76433-7>

Library of Congress Control Number: 2018936916

© Springer International Publishing AG, part of Springer Nature 2018

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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature.

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Sally, Bridget, Jean, Yeshe, Nyima, and
Lhamo.*

Preface

The minimum message length (MML) principle was devised by Chris Wallace (1933–2004) and David Boulton in the late 1960s [12, 93] initially to solve the unsupervised mixture modelling problem—an important problem, a mathematical analysis, and a working computer program (Snob) that gives useful results in many different areas of science, a “complete” research project.

The Foundation Chair of Computer Science at Monash University, Chris is also particularly remembered for his work on the “Wallace multiplier” [85, 86], pseudo-random number generators [14, 89], and operating systems [6, 99].

MML was developed [91, 92] in practical and theoretical directions and was applied to many inference problems by Chris, co-workers, postgraduates, and postdocs. One of my personal favourite applications is Jon Patrick’s modelling of megalithic stone circles [65, 66].

I first heard about MML over lunch one day which led to applying it to biological sequence alignment [3] and related problems [15], and eventually after many twists and turns to protein structural alignment [17] and protein folding patterns [83].

Unfortunately much MML-based research that led to new inductive inference programs resulted in little shared software *componentry*. A new program tended to be written largely from scratch by a postgrad, postdoc or other researcher and did not contribute to any software library of shared parts. As such the programs embody reimplementations of standard parts. This phenomenon is not due to any special property of MML and actually seems to be quite common in research but it is rather ironic because, what with the complexity of models and of data being measured in the same units, MML is well suited to writing components that can be reused and supplied as parameters to other inductive inference software.

The first MML book is the one written by Chris Wallace [92] but published posthumously; it is the reference work for MML theory. This other MML book is an attempt to do a combined MML and Software Engineering analysis of inductive inference software. Sound programming skills are needed to write new application

programs for inductive inference problems. Some mathematical skills, particularly in calculus and linear algebra, are needed to do a new MML analysis of one's favourite statistical model.

Melbourne, Victoria, Australia

Lloyd Allison

Acknowledgements

Chris Wallace was a great inspiration and always generous with ideas. He is sadly missed. This book was begun largely at the urging of Arun Konagurthu who at times shows aspects of both an irresistible force and an immovable object. He also contributed to the content and examples and fought valiantly in the typesetting wars.

Leigh Fitzgibbon, Josh Comley, and Rodney O'Donnell deserve special mention for contributions [18, 34, 61] to early attempts to create and use *general* MML software.

Many thanks go to Dianna Kenny for sharing her data on musicians and mortality (Sect. 7.5).

I am indebted to those who read parts, a little or a lot, of drafts of the book and who suggested improvements, a few or many, in alphabetical order: Rohan Baxter, Minh Duc Cao, Trevor Dix, Rodney O'Donnell, Arun Konagurthu, Francois Petitjean, Joel Reicher, Daniel Schmidt. But, as they say, the mistakes are all my own.

Contents

Preface	vii
Acknowledgements	ix
1 Introduction	1
1.1 Explanation versus Prediction	4
1.2 Models.....	5
1.2.1 Implementation	5
1.3 Estimators	6
1.4 Information.....	8
1.5 MML	9
1.6 MML87.....	10
1.6.1 Single Continuous Parameter θ	10
1.6.2 Multiple Continuous Parameters	13
1.7 Outline.....	14
2 Discrete	17
2.1 Uniform	19
2.2 Two State	19
2.3 MultiState	21
2.4 Adaptive	23
3 Integers	27
3.1 Universal Codes.....	28
3.2 Wallace Tree Code	30
3.3 A Warning.....	32
3.4 Geometric	32
3.5 Poisson	36
3.6 Shifting	38

4	Continuous	41
4.1	Uniform	42
4.2	Exponential	43
4.3	Normal	45
4.4	Normal Given μ	48
4.5	Laplace	48
4.6	Comparing Models	52
5	Function-Models	53
5.1	Konstant Function-Model, K	54
5.2	Multinomial	55
5.3	Intervals	56
5.4	Conditional Probability Table (CPT)	57
6	Multivariate	61
6.1	Independent	62
6.2	Dependent	63
6.3	Data Operations	64
7	Mixture Models	65
7.1	The Message	66
7.2	Search	69
7.3	Implementation	70
7.4	An Example Mixture Model	71
7.5	The 27 Club	74
8	Function-Models 2	77
8.1	Naive Bayes	78
8.2	An Example	79
8.3	Note: Not So Naive	80
8.4	Classification Trees	81
8.5	The Message	82
8.6	Search	84
8.7	Implementation	85
8.8	An Example	86
8.9	Missing Values	88
9	Vectors	89
9.1	D -Dimensions, \mathbb{R}^D	90
9.1.1	Implementation	90
9.1.2	Norm and Direction	91

9.2	Simplex and $\ \vec{x}\ _1 = 1$	92
9.2.1	Uniform	93
9.2.2	Implementation	93
9.2.3	Dirichlet	94
9.3	Directions in \mathbb{R}^D	96
9.3.1	Uniform	97
9.3.2	von Mises–Fisher (vMF) Distribution	97
10	Linear Regression	103
10.1	Single Variable	103
10.1.1	Implementation	106
10.2	Single Dependence	107
10.2.1	Unknown Single Dependence	109
10.3	Multiple Input Variables	109
10.3.1	Implementation	111
11	Graphs	113
11.1	Ordered or Unordered Graphs	114
11.2	Adjacency Matrix v. Adjacency Lists	117
11.3	Models of Graphs	117
11.4	Gilbert, Erdos and Renyi Models	117
11.5	Gilbert, Erdos and Renyi Adaptive	119
11.6	Skewed Degree Model	120
11.7	Motif Models	121
11.8	A Simple Motif Model	122
11.9	An Adaptive Motif Model	124
11.10	Comparisons	126
11.11	Biological Networks	127
12	Bits and Pieces	131
12.1	Priors	132
12.2	Parameterisation	132
12.3	Estimators	133
12.4	Data Accuracy of Measurement (AoM)	134
12.5	Small or Big Data	134
12.6	Data Compression Techniques	135
12.7	Getting Started	136
12.8	Testing and Evaluating Results	137
12.8.1	Evaluating Function-Models	138
12.9	Programming Languages	138
12.10	logSum	139
12.11	Data-Sets	140

13	An Implementation	143
13.1	Support	143
13.2	Values.....	144
13.3	Utilities	146
13.4	Maths	146
13.5	Vectors.....	147
13.6	Graphs	148
13.7	Models.....	149
13.8	Example Programs	153
	Glossary	155
	Bibliography	167
	Index.....	173