

Ian H. Witten • Eibe Frank • Mark A. Hall

DATA MINING

Practical Machine Learning Tools and Techniques

THIRD EDITION

MK
MORGAN KAUFMANN



Data Mining

Third Edition

Data Mining

Practical Machine Learning Tools and Techniques

Third Edition

Ian H. Witten

Eibe Frank

Mark A. Hall



AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann Publishers is an imprint of Elsevier



Morgan Kaufmann Publishers is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

This book is printed on acid-free paper.

Copyright © 2011 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

Witten, I. H. (Ian H.)

Data mining : practical machine learning tools and techniques.—3rd ed. /

Ian H. Witten, Frank Eibe, Mark A. Hall.

p. cm.—(The Morgan Kaufmann series in data management systems)

ISBN 978-0-12-374856-0 (pbk.)

1. Data mining. I. Hall, Mark A. II. Title.

QA76.9.D343W58 2011

006.3'12—dc22

2010039827

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

For information on all Morgan Kaufmann publications, visit our
website at www.mkp.com or www.elsevierdirect.com

Printed in the United States

11 12 13 14 15 10 9 8 7 6 5 4 3 2 1

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER

BOOK AID
International

Sabre Foundation

Contents

LIST OF FIGURES	xv
LIST OF TABLES	xix
PREFACE	xxi
Updated and Revised Content	xxv
Second Edition	xxv
Third Edition	xxvi
ACKNOWLEDGMENTS	xxix
ABOUT THE AUTHORS	xxxiii

PART I INTRODUCTION TO DATA MINING

CHAPTER 1	What's It All About?	3
1.1	Data Mining and Machine Learning	3
	Describing Structural Patterns	5
	Machine Learning	7
	Data Mining	8
1.2	Simple Examples: The Weather Problem and Others	9
	The Weather Problem	9
	Contact Lenses: An Idealized Problem	12
	Iris: A Classic Numeric Dataset	13
	CPU Performance: Introducing Numeric Prediction	15
	Labor Negotiations: A More Realistic Example	15
	Soybean Classification: A Classic Machine Learning Success	19
1.3	Fielded Applications	21
	Web Mining	21
	Decisions Involving Judgment	22
	Screening Images	23
	Load Forecasting	24
	Diagnosis	25
	Marketing and Sales	26
	Other Applications	27
1.4	Machine Learning and Statistics	28
1.5	Generalization as Search	29
1.6	Data Mining and Ethics	33
	Reidentification	33
	Using Personal Information	34
	Wider Issues	35
1.7	Further Reading	36

CHAPTER 2	Input: Concepts, Instances, and Attributes	39
2.1	What's a Concept?	40
2.2	What's in an Example?	42
	Relations.....	43
	Other Example Types.....	46
2.3	What's in an Attribute?	49
2.4	Preparing the Input	51
	Gathering the Data Together.....	51
	ARFF Format	52
	Sparse Data	56
	Attribute Types.....	56
	Missing Values	58
	Inaccurate Values	59
	Getting to Know Your Data.....	60
2.5	Further Reading	60
CHAPTER 3	Output: Knowledge Representation	61
3.1	Tables	61
3.2	Linear Models	62
3.3	Trees	64
3.4	Rules.....	67
	Classification Rules.....	69
	Association Rules.....	72
	Rules with Exceptions	73
	More Expressive Rules	75
3.5	Instance-Based Representation	78
3.6	Clusters.....	81
3.7	Further Reading	83
CHAPTER 4	Algorithms: The Basic Methods	85
4.1	Inferring Rudimentary Rules	86
	Missing Values and Numeric Attributes	87
	Discussion	89
4.2	Statistical Modeling	90
	Missing Values and Numeric Attributes	94
	Naïve Bayes for Document Classification.....	97
	Discussion	99
4.3	Divide-and-Conquer: Constructing Decision Trees	99
	Calculating Information	103
	Highly Branching Attributes.....	105
	Discussion	107

4.4	Covering Algorithms: Constructing Rules	108
	Rules versus Trees	109
	A Simple Covering Algorithm.....	110
	Rules versus Decision Lists.....	115
4.5	Mining Association Rules.....	116
	Item Sets.....	116
	Association Rules.....	119
	Generating Rules Efficiently.....	122
	Discussion	123
4.6	Linear Models	124
	Numeric Prediction: Linear Regression	124
	Linear Classification: Logistic Regression.....	125
	Linear Classification Using the Perceptron.....	127
	Linear Classification Using Winnow.....	129
4.7	Instance-Based Learning.....	131
	Distance Function	131
	Finding Nearest Neighbors Efficiently	132
	Discussion	137
4.8	Clustering	138
	Iterative Distance-Based Clustering	139
	Faster Distance Calculations.....	139
	Discussion	141
4.9	Multi-Instance Learning.....	141
	Aggregating the Input.....	142
	Aggregating the Output	142
	Discussion	142
4.10	Further Reading	143
4.11	Weka Implementations.....	145
CHAPTER 5	Credibility: Evaluating What's Been Learned.....	147
5.1	Training and Testing	148
5.2	Predicting Performance.....	150
5.3	Cross-Validation.....	152
5.4	Other Estimates	154
	Leave-One-Out Cross-Validation.....	154
	The Bootstrap.....	155
5.5	Comparing Data Mining Schemes.....	156
5.6	Predicting Probabilities.....	159
	Quadratic Loss Function.....	160
	Informational Loss Function.....	161
	Discussion	162

5.7	Counting the Cost	163
	Cost-Sensitive Classification	166
	Cost-Sensitive Learning.....	167
	Lift Charts	168
	ROC Curves	172
	Recall–Precision Curves	174
	Discussion	175
	Cost Curves	177
5.8	Evaluating Numeric Prediction.....	180
5.9	Minimum Description Length Principle.....	183
5.10	Applying the MDL Principle to Clustering.....	186
5.11	Further Reading	187

PART II ADVANCED DATA MINING

CHAPTER 6	Implementations: Real Machine Learning Schemes.....	191
6.1	Decision Trees.....	192
	Numeric Attributes.....	193
	Missing Values	194
	Pruning	195
	Estimating Error Rates.....	197
	Complexity of Decision Tree Induction	199
	From Trees to Rules.....	200
	C4.5: Choices and Options	201
	Cost-Complexity Pruning	202
	Discussion	202
6.2	Classification Rules.....	203
	Criteria for Choosing Tests.....	203
	Missing Values, Numeric Attributes.....	204
	Generating Good Rules.....	205
	Using Global Optimization.....	208
	Obtaining Rules from Partial Decision Trees.....	208
	Rules with Exceptions	212
	Discussion	215
6.3	Association Rules.....	216
	Building a Frequent-Pattern Tree	216
	Finding Large Item Sets	219
	Discussion	222
6.4	Extending Linear Models	223
	Maximum-Margin Hyperplane	224
	Nonlinear Class Boundaries	226

	Support Vector Regression.....	227
	Kernel Ridge Regression	229
	Kernel Perceptron	231
	Multilayer Perceptrons.....	232
	Radial Basis Function Networks	241
	Stochastic Gradient Descent.....	242
	Discussion	243
6.5	Instance-Based Learning.....	244
	Reducing the Number of Exemplars	245
	Pruning Noisy Exemplars.....	245
	Weighting Attributes	246
	Generalizing Exemplars.....	247
	Distance Functions for Generalized Exemplars.....	248
	Generalized Distance Functions	249
	Discussion	250
6.6	Numeric Prediction with Local Linear Models.....	251
	Model Trees	252
	Building the Tree	253
	Pruning the Tree.....	253
	Nominal Attributes.....	254
	Missing Values	254
	Pseudocode for Model Tree Induction	255
	Rules from Model Trees	259
	Locally Weighted Linear Regression.....	259
	Discussion	261
6.7	Bayesian Networks	261
	Making Predictions	262
	Learning Bayesian Networks.....	266
	Specific Algorithms.....	268
	Data Structures for Fast Learning	270
	Discussion	273
6.8	Clustering	273
	Choosing the Number of Clusters	274
	Hierarchical Clustering	274
	Example of Hierarchical Clustering	276
	Incremental Clustering.....	279
	Category Utility	284
	Probability-Based Clustering	285
	The EM Algorithm.....	287
	Extending the Mixture Model	289

	Bayesian Clustering	290
	Discussion	292
6.9	Semisupervised Learning.....	294
	Clustering for Classification	294
	Co-training	296
	EM and Co-training	297
	Discussion	297
6.10	Multi-Instance Learning.....	298
	Converting to Single-Instance Learning.....	298
	Upgrading Learning Algorithms.....	300
	Dedicated Multi-Instance Methods.....	301
	Discussion	302
6.11	Weka Implementations.....	303
CHAPTER 7	Data Transformations	305
7.1	Attribute Selection	307
	Scheme-Independent Selection.....	308
	Searching the Attribute Space	311
	Scheme-Specific Selection.....	312
7.2	Discretizing Numeric Attributes	314
	Unsupervised Discretization	316
	Entropy-Based Discretization	316
	Other Discretization Methods.....	320
	Entropy-Based versus Error-Based Discretization.....	320
	Converting Discrete Attributes to Numeric Attributes.....	322
7.3	Projections.....	322
	Principal Components Analysis.....	324
	Random Projections	326
	Partial Least-Squares Regression	326
	Text to Attribute Vectors.....	328
	Time Series	330
7.4	Sampling	330
	Reservoir Sampling.....	330
7.5	Cleansing.....	331
	Improving Decision Trees.....	332
	Robust Regression	333
	Detecting Anomalies.....	334
	One-Class Learning	335
7.6	Transforming Multiple Classes to Binary Ones.....	338
	Simple Methods	338
	Error-Correcting Output Codes	339
	Ensembles of Nested Dichotomies.....	341

7.7	Calibrating Class Probabilities	343
7.8	Further Reading	346
7.9	Weka Implementations.....	348
CHAPTER 8	Ensemble Learning	351
8.1	Combining Multiple Models.....	351
8.2	Bagging	352
	Bias–Variance Decomposition	353
	Bagging with Costs.....	355
8.3	Randomization	356
	Randomization versus Bagging	357
	Rotation Forests	357
8.4	Boosting	358
	AdaBoost.....	358
	The Power of Boosting.....	361
8.5	Additive Regression.....	362
	Numeric Prediction	362
	Additive Logistic Regression	364
8.6	Interpretable Ensembles.....	365
	Option Trees.....	365
	Logistic Model Trees	368
8.7	Stacking.....	369
8.8	Further Reading	371
8.9	Weka Implementations.....	372
Chapter 9	Moving on: Applications and Beyond.....	375
9.1	Applying Data Mining.....	375
9.2	Learning from Massive Datasets	378
9.3	Data Stream Learning	380
9.4	Incorporating Domain Knowledge	384
9.5	Text Mining.....	386
9.6	Web Mining.....	389
9.7	Adversarial Situations.....	393
9.8	Ubiquitous Data Mining	395
9.9	Further Reading	397

PART III THE WEKA DATA MINING WORKBENCH

CHAPTER 10	Introduction to Weka	403
10.1	What’s in Weka?	403
10.2	How Do You Use It?	404
10.3	What Else Can You Do?	405
10.4	How Do You Get It?	406

CHAPTER 11	The Explorer	407
11.1	Getting Started	407
	Preparing the Data	407
	Loading the Data into the Explorer	408
	Building a Decision Tree	410
	Examining the Output	411
	Doing It Again	413
	Working with Models	414
	When Things Go Wrong	415
11.2	Exploring the Explorer	416
	Loading and Filtering Files	416
	Training and Testing Learning Schemes	422
	Do It Yourself: The User Classifier	424
	Using a Metalearner	427
	Clustering and Association Rules	429
	Attribute Selection	430
	Visualization	430
11.3	Filtering Algorithms	432
	Unsupervised Attribute Filters	432
	Unsupervised Instance Filters	441
	Supervised Filters	443
11.4	Learning Algorithms	445
	Bayesian Classifiers	451
	Trees	454
	Rules	457
	Functions	459
	Neural Networks	469
	Lazy Classifiers	472
	Multi-Instance Classifiers	472
	Miscellaneous Classifiers	474
11.5	Metalearning Algorithms	474
	Bagging and Randomization	474
	Boosting	476
	Combining Classifiers	477
	Cost-Sensitive Learning	477
	Optimizing Performance	478
	Retargeting Classifiers for Different Tasks	479
11.6	Clustering Algorithms	480
11.7	Association-Rule Learners	485
11.8	Attribute Selection	487
	Attribute Subset Evaluators	488

Single-Attribute Evaluators	490
Search Methods.....	492
CHAPTER 12 The Knowledge Flow Interface	495
12.1 Getting Started	495
12.2 Components.....	498
12.3 Configuring and Connecting the Components	500
12.4 Incremental Learning	502
CHAPTER 13 The Experimenter	505
13.1 Getting Started	505
Running an Experiment	506
Analyzing the Results	509
13.2 Simple Setup	510
13.3 Advanced Setup	511
13.4 The Analyze Panel	512
13.5 Distributing Processing over Several Machines.....	515
CHAPTER 14 The Command-Line Interface.....	519
14.1 Getting Started	519
14.2 The Structure of Weka.....	519
Classes, Instances, and Packages.....	520
The <i>weka.core</i> Package.....	520
The <i>weka.classifiers</i> Package.....	523
Other Packages.....	525
Javadoc Indexes	525
14.3 Command-Line Options.....	526
Generic Options	526
Scheme-Specific Options	529
CHAPTER 15 Embedded Machine Learning	531
15.1 A Simple Data Mining Application	531
<i>MessageClassifier()</i>	536
<i>updateData()</i>	536
<i>classifyMessage()</i>	537
CHAPTER 16 Writing New Learning Schemes	539
16.1 An Example Classifier	539
<i>buildClassifier()</i>	540
<i>makeTree()</i>	540
<i>computeInfoGain()</i>	549
<i>classifyInstance()</i>	549

<i>toSource()</i>	550
<i>main()</i>	553
16.2 Conventions for Implementing Classifiers	555
Capabilities	555
CHAPTER 17 Tutorial Exercises for the Weka Explorer.....	559
17.1 Introduction to the Explorer Interface	559
Loading a Dataset	559
The Dataset Editor	560
Applying a Filter	561
The Visualize Panel	562
The Classify Panel	562
17.2 Nearest-Neighbor Learning and Decision Trees	566
The Glass Dataset	566
Attribute Selection	567
Class Noise and Nearest-Neighbor Learning	568
Varying the Amount of Training Data	569
Interactive Decision Tree Construction	569
17.3 Classification Boundaries	571
Visualizing 1R	571
Visualizing Nearest-Neighbor Learning	572
Visualizing Naïve Bayes	573
Visualizing Decision Trees and Rule Sets	573
Messing with the Data	574
17.4 Preprocessing and Parameter Tuning	574
Discretization	574
More on Discretization	575
Automatic Attribute Selection	575
More on Automatic Attribute Selection	576
Automatic Parameter Tuning	577
17.5 Document Classification	578
Data with String Attributes	579
Classifying Actual Documents	580
Exploring the <i>StringToWordVector</i> Filter	581
17.6 Mining Association Rules	582
Association-Rule Mining	582
Mining a Real-World Dataset	584
Market Basket Analysis	584
REFERENCES.....	587
INDEX.....	607

List of Figures

Figure 1.1 Rules for the contact lens data.	12
Figure 1.2 Decision tree for the contact lens data.	13
Figure 1.3 Decision trees for the labor negotiations data.	18
Figure 2.1 A family tree and two ways of expressing the sister-of relation.	43
Figure 2.2 ARFF file for the weather data.	53
Figure 2.3 Multi-instance ARFF file for the weather data.	55
Figure 3.1 A linear regression function for the CPU performance data.	62
Figure 3.2 A linear decision boundary separating <i>Iris setosas</i> from <i>Iris versicolors</i> .	63
Figure 3.3 Constructing a decision tree interactively.	66
Figure 3.4 Models for the CPU performance data.	68
Figure 3.5 Decision tree for a simple disjunction.	69
Figure 3.6 The exclusive-or problem.	70
Figure 3.7 Decision tree with a replicated subtree.	71
Figure 3.8 Rules for the iris data.	74
Figure 3.9 The shapes problem.	76
Figure 3.10 Different ways of partitioning the instance space.	80
Figure 3.11 Different ways of representing clusters.	82
Figure 4.1 Pseudocode for 1R.	86
Figure 4.2 Tree stumps for the weather data.	100
Figure 4.3 Expanded tree stumps for the weather data.	102
Figure 4.4 Decision tree for the weather data.	103
Figure 4.5 Tree stump for the <i>ID code</i> attribute.	105
Figure 4.6 Covering algorithm.	109
Figure 4.7 The instance space during operation of a covering algorithm.	110
Figure 4.8 Pseudocode for a basic rule learner.	114
Figure 4.9 Logistic regression.	127
Figure 4.10 The perceptron.	129
Figure 4.11 The Winnow algorithm.	130
Figure 4.12 A <i>kD</i> -tree for four training instances.	133
Figure 4.13 Using a <i>kD</i> -tree to find the nearest neighbor of the star.	134
Figure 4.14 Ball tree for 16 training instances.	136
Figure 4.15 Ruling out an entire ball (gray) based on a target point (star) and its current nearest neighbor.	137
Figure 4.16 A ball tree.	141
Figure 5.1 A hypothetical lift chart.	170
Figure 5.2 Analyzing the expected benefit of a mailing campaign.	171
Figure 5.3 A sample ROC curve.	173
Figure 5.4 ROC curves for two learning schemes.	174
Figure 5.5 Effect of varying the probability threshold.	178
Figure 6.1 Example of subtree raising.	196

Figure 6.2 Pruning the labor negotiations decision tree.	200
Figure 6.3 Algorithm for forming rules by incremental reduced-error pruning.	207
Figure 6.4 RIPPER.	209
Figure 6.5 Algorithm for expanding examples into a partial tree.	210
Figure 6.6 Example of building a partial tree.	211
Figure 6.7 Rules with exceptions for the iris data.	213
Figure 6.8 Extended prefix trees for the weather data.	220
Figure 6.9 A maximum-margin hyperplane.	225
Figure 6.10 Support vector regression.	228
Figure 6.11 Example datasets and corresponding perceptrons.	233
Figure 6.12 Step versus sigmoid.	240
Figure 6.13 Gradient descent using the error function $w^2 + 1$.	240
Figure 6.14 Multilayer perceptron with a hidden layer.	241
Figure 6.15 Hinge, squared, and 0 – 1 loss functions.	242
Figure 6.16 A boundary between two rectangular classes.	248
Figure 6.17 Pseudocode for model tree induction.	255
Figure 6.18 Model tree for a dataset with nominal attributes.	256
Figure 6.19 A simple Bayesian network for the weather data.	262
Figure 6.20 Another Bayesian network for the weather data.	264
Figure 6.21 The weather data.	270
Figure 6.22 Hierarchical clustering displays.	276
Figure 6.23 Clustering the weather data.	279
Figure 6.24 Hierarchical clusterings of the iris data.	281
Figure 6.25 A two-class mixture model.	285
Figure 6.26 <i>DensiTree</i> showing possible hierarchical clusterings of a given dataset.	291
Figure 7.1 Attribute space for the weather dataset.	311
Figure 7.2 Discretizing the <i>temperature</i> attribute using the entropy method.	318
Figure 7.3 The result of discretizing the <i>temperature</i> attribute.	318
Figure 7.4 Class distribution for a two-class, two-attribute problem.	321
Figure 7.5 Principal components transform of a dataset.	325
Figure 7.6 Number of international phone calls from Belgium, 1950–1973.	333
Figure 7.7 Overoptimistic probability estimation for a two-class problem.	344
Figure 8.1 Algorithm for bagging.	355
Figure 8.2 Algorithm for boosting.	359
Figure 8.3 Algorithm for additive logistic regression.	365
Figure 8.4 Simple option tree for the weather data.	366
Figure 8.5 Alternating decision tree for the weather data.	367
Figure 9.1 A tangled “web.”	391
Figure 11.1 The Explorer interface.	408
Figure 11.2 Weather data.	409
Figure 11.3 The Weka Explorer.	410

Figure 11.4 Using <i>J4.8</i> .	411
Figure 11.5 Output from the <i>J4.8</i> decision tree learner.	412
Figure 11.6 Visualizing the result of <i>J4.8</i> on the iris dataset.	415
Figure 11.7 Generic Object Editor.	417
Figure 11.8 The <i>SQLViewer</i> tool.	418
Figure 11.9 Choosing a filter.	420
Figure 11.10 The weather data with two attributes removed.	422
Figure 11.11 Processing the CPU performance data with <i>M5'</i> .	423
Figure 11.12 Output from the <i>M5'</i> program for numeric prediction.	425
Figure 11.13 Visualizing the errors.	426
Figure 11.14 Working on the segment-challenge data with the User Classifier.	428
Figure 11.15 Configuring a metalearner for boosting decision stumps.	429
Figure 11.16 Output from the <i>Apriori</i> program for association rules.	430
Figure 11.17 Visualizing the iris dataset.	431
Figure 11.18 Using Weka's metalearner for discretization.	443
Figure 11.19 Output of <i>NaiveBayes</i> on the weather data.	452
Figure 11.20 Visualizing a Bayesian network for the weather data (nominal version).	454
Figure 11.21 Changing the parameters for <i>J4.8</i> .	455
Figure 11.22 Output of <i>OneR</i> on the labor negotiations data.	458
Figure 11.23 Output of <i>PART</i> for the labor negotiations data.	460
Figure 11.24 Output of <i>SimpleLinearRegression</i> for the CPU performance data.	461
Figure 11.25 Output of <i>SMO</i> on the iris data.	463
Figure 11.26 Output of <i>SMO</i> with a nonlinear kernel on the iris data.	465
Figure 11.27 Output of <i>Logistic</i> on the iris data.	468
Figure 11.28 Using Weka's neural-network graphical user interface.	470
Figure 11.29 Output of <i>SimpleKMeans</i> on the weather data.	481
Figure 11.30 Output of <i>EM</i> on the weather data.	482
Figure 11.31 Clusters formed by <i>DBScan</i> on the iris data.	484
Figure 11.32 <i>OPTICS</i> visualization for the iris data.	485
Figure 11.33 Attribute selection: specifying an evaluator and a search method.	488
Figure 12.1 The Knowledge Flow interface.	496
Figure 12.2 Configuring a data source.	497
Figure 12.3 Status area after executing the configuration shown in Figure 12.1.	497
Figure 12.4 Operations on the Knowledge Flow components.	500
Figure 12.5 A Knowledge Flow that operates incrementally.	503
Figure 13.1 An experiment.	506
Figure 13.2 Statistical test results for the experiment in Figure 13.1.	509
Figure 13.3 Setting up an experiment in advanced mode.	511
Figure 13.4 An experiment in clustering.	513

Figure 13.5 Rows and columns of Figure 13.2.	514
Figure 14.1 Using Javadoc.	521
Figure 14.2 <i>DecisionStump</i> , a class of the <i>weka.classifiers.trees</i> package.	524
Figure 15.1 Source code for the message classifier.	532
Figure 16.1 Source code for the ID3 decision tree learner.	541
Figure 16.2 Source code produced by <i>weka.classifiers.trees.Id3</i> for the weather data.	551
Figure 16.3 Javadoc for the <i>Capability</i> enumeration.	556
Figure 17.1 The data viewer.	560
Figure 17.2 Output after building and testing the classifier.	564
Figure 17.3 The decision tree that has been built.	565

List of Tables

Table 1.1 Contact Lens Data	6
Table 1.2 Weather Data	10
Table 1.3 Weather Data with Some Numeric Attributes	11
Table 1.4 Iris Data	14
Table 1.5 CPU Performance Data	16
Table 1.6 Labor Negotiations Data	17
Table 1.7 Soybean Data	20
Table 2.1 Iris Data as a Clustering Problem	41
Table 2.2 Weather Data with a Numeric Class	42
Table 2.3 Family Tree	44
Table 2.4 Sister-of Relation	45
Table 2.5 Another Relation	47
Table 3.1 New Iris Flower	73
Table 3.2 Training Data for the Shapes Problem	76
Table 4.1 Evaluating Attributes in the Weather Data	87
Table 4.2 Weather Data with Counts and Probabilities	91
Table 4.3 A New Day	92
Table 4.4 Numeric Weather Data with Summary Statistics	95
Table 4.5 Another New Day	96
Table 4.6 Weather Data with Identification Codes	106
Table 4.7 Gain Ratio Calculations for Figure 4.2 Tree Stumps	107
Table 4.8 Part of Contact Lens Data for which <i>astigmatism</i> = <i>yes</i>	112
Table 4.9 Part of Contact Lens Data for which <i>astigmatism</i> = <i>yes</i> and <i>tear production rate</i> = <i>normal</i>	113
Table 4.10 Item Sets for Weather Data with Coverage 2 or Greater	117
Table 4.11 Association Rules for Weather Data	120
Table 5.1 Confidence Limits for Normal Distribution	152
Table 5.2 Confidence Limits for Student's Distribution with 9 Degrees of Freedom	159
Table 5.3 Different Outcomes of a Two-Class Prediction	164
Table 5.4 Different Outcomes of a Three-Class Prediction	165
Table 5.5 Default Cost Matrixes	166
Table 5.6 Data for a Lift Chart	169
Table 5.7 Different Measures Used to Evaluate the False Positive versus False Negative Trade-Off	176
Table 5.8 Performance Measures for Numeric Prediction	180
Table 5.9 Performance Measures for Four Numeric Prediction Models	182
Table 6.1 Preparing Weather Data for Insertion into an <i>FP</i> -Tree	217
Table 6.2 Linear Models in the Model Tree	257
Table 7.1 First Five Instances from CPU Performance Data	327
Table 7.2 Transforming a Multiclass Problem into a Two-Class One	340

Table 7.3 Nested Dichotomy in the Form of a Code Matrix	342
Table 9.1 Top 10 Algorithms in Data Mining	376
Table 11.1 Unsupervised Attribute Filters	433
Table 11.2 Unsupervised Instance Filters	441
Table 11.3 Supervised Attribute Filters	444
Table 11.4 Supervised Instance Filters	444
Table 11.5 Classifier Algorithms in Weka	446
Table 11.6 Metalearning Algorithms in Weka	475
Table 11.7 Clustering Algorithms	480
Table 11.8 Association-Rule Learners	486
Table 11.9 Attribute Evaluation Methods for Attribute Selection	489
Table 11.10 Search Methods for Attribute Selection	490
Table 12.1 Visualization and Evaluation Components	499
Table 14.1 Generic Options for Learning Schemes	527
Table 14.2 Scheme-Specific Options for the J4.8 Decision Tree Learner	528
Table 16.1 Simple Learning Schemes in Weka	540
Table 17.1 Accuracy Obtained Using <i>IBk</i> , for Different Attribute Subsets	568
Table 17.2 Effect of Class Noise on <i>IBk</i> , for Different Neighborhood Sizes	569
Table 17.3 Effect of Training Set Size on <i>IBk</i> and <i>J48</i>	570
Table 17.4 Training Documents	580
Table 17.5 Test Documents	580
Table 17.6 Number of Rules for Different Values of Minimum Confidence and Support	584

Preface

The convergence of computing and communication has produced a society that feeds on information. Yet most of the information is in its raw form: data. If *data* is characterized as recorded facts, then *information* is the set of patterns, or expectations, that underlie the data. There is a huge amount of information locked up in databases—information that is potentially important but has not yet been discovered or articulated. Our mission is to bring it forth.

Data mining is the extraction of implicit, previously unknown, and potentially useful information from data. The idea is to build computer programs that sift through databases automatically, seeking regularities or patterns. Strong patterns, if found, will likely generalize to make accurate predictions on future data. Of course, there will be problems. Many patterns will be banal and uninteresting. Others will be spurious, contingent on accidental coincidences in the particular dataset used. And real data is imperfect: Some parts will be garbled, some missing. Anything that is discovered will be inexact: There will be exceptions to every rule and cases not covered by any rule. Algorithms need to be robust enough to cope with imperfect data and to extract regularities that are inexact but useful.

Machine learning provides the technical basis of data mining. It is used to extract information from the raw data in databases—information that is expressed in a comprehensible form and can be used for a variety of purposes. The process is one of abstraction: taking the data, warts and all, and inferring whatever structure underlies it. This book is about the tools and techniques of machine learning that are used in practical data mining for finding, and describing, structural patterns in data.

As with any burgeoning new technology that enjoys intense commercial attention, the use of data mining is surrounded by a great deal of hype in the technical—and sometimes the popular—press. Exaggerated reports appear of the secrets that can be uncovered by setting learning algorithms loose on oceans of data. But there is no magic in machine learning, no hidden power, no alchemy. Instead, there is an identifiable body of simple and practical techniques that can often extract useful information from raw data. This book describes these techniques and shows how they work.

We interpret machine learning as the acquisition of structural descriptions from examples. The kind of descriptions that are found can be used for prediction, explanation, and understanding. Some data mining applications focus on prediction: They forecast what will happen in new situations from data that describe what happened in the past, often by guessing the classification of new examples. But we are equally—perhaps more—interested in applications where the result of “learning” is an actual description of a structure that can be used to classify examples. This structural description supports explanation and understanding as well as prediction. In our experience, insights gained by the user are of most interest in the majority of practical data mining applications; indeed, this is one of machine learning’s major advantages over classical statistical modeling.

The book explains a wide variety of machine learning methods. Some are pedagogically motivated: simple schemes that are designed to explain clearly how the basic ideas work. Others are practical: real systems that are used in applications today. Many are contemporary and have been developed only in the last few years.

A comprehensive software resource has been created to illustrate the ideas in this book. Called the Waikato Environment for Knowledge Analysis, or Weka¹ for short, it is available as Java source code at www.cs.waikato.ac.nz/ml/weka. It is a full, industrial-strength implementation of essentially all the techniques that are covered in this book. It includes illustrative code and working implementations of machine learning methods. It offers clean, spare implementations of the simplest techniques, designed to aid understanding of the mechanisms involved. It also provides a workbench that includes full, working, state-of-the-art implementations of many popular learning schemes that can be used for practical data mining or for research. Finally, it contains a framework, in the form of a Java class library, that supports applications that use embedded machine learning and even the implementation of new learning schemes.

The objective of this book is to introduce the tools and techniques for machine learning that are used in data mining. After reading it, you will understand what these techniques are and appreciate their strengths and applicability. If you wish to experiment with your own data, you will be able to do this easily with the Weka software.

The book spans the gulf between the intensely practical approach taken by trade books that provide case studies on data mining and the more theoretical, principle-driven exposition found in current textbooks on machine learning. (A brief description of these books appears in the Further Reading section at the end of Chapter 1.) This gulf is rather wide. To apply machine learning techniques productively, you need to understand something about how they work; this is not a technology that you can apply blindly and expect to get good results. Different problems yield to different techniques, but it is rarely obvious which techniques are suitable for a given situation: You need to know something about the range of possible solutions. And we cover an extremely wide range of techniques. We can do this because, unlike many trade books, this volume does not promote any particular commercial software or approach. We include a large number of examples, but they use illustrative datasets that are small enough to allow you to follow what is going on. Real datasets are far too large to show this (and in any case are usually company confidential). Our datasets are chosen not to illustrate actual large-scale practical problems but to help you understand what the different techniques do, how they work, and what their range of application is.

The book is aimed at the technically aware general reader who is interested in the principles and ideas underlying the current practice of data mining. It will also

¹Found only on the islands of New Zealand, the weka (pronounced to rhyme with “Mecca”) is a flightless bird with an inquisitive nature.

be of interest to information professionals who need to become acquainted with this new technology, and to all those who wish to gain a detailed technical understanding of what machine learning involves. It is written for an eclectic audience of information systems practitioners, programmers, consultants, developers, information technology managers, specification writers, patent examiners, and curious lay people, as well as students and professors, who need an easy-to-read book with lots of illustrations that describes what the major machine learning techniques are, what they do, how they are used, and how they work. It is practically oriented, with a strong “how to” flavor, and includes algorithms, code, and implementations. All those involved in practical data mining will benefit directly from the techniques described. The book is aimed at people who want to cut through to the reality that underlies the hype about machine learning and who seek a practical, nonacademic, unpretentious approach. We have avoided requiring any specific theoretical or mathematical knowledge, except in some sections that are marked by a box around the text. These contain optional material, often for the more technically or theoretically inclined reader, and may be skipped without loss of continuity.

The book is organized in layers that make the ideas accessible to readers who are interested in grasping the basics, as well as accessible to those who would like more depth of treatment, along with full details on the techniques covered. We believe that consumers of machine learning need to have some idea of how the algorithms they use work. It is often observed that data models are only as good as the person who interprets them, and that person needs to know something about how the models are produced to appreciate the strengths, and limitations, of the technology. However, it is not necessary for all users to have a deep understanding of the finer details of the algorithms.

We address this situation by describing machine learning methods at successive levels of detail. The book is divided into three parts. Part I is an introduction to data mining. The reader will learn the basic ideas, the topmost level, by reading the first three chapters. Chapter 1 describes, through examples, what machine learning is and where it can be used; it also provides actual practical applications. Chapters 2 and 3 cover the different kinds of input and output, or *knowledge representation*, that are involved—different kinds of output dictate different styles of algorithm. Chapter 4 describes the basic methods of machine learning, simplified to make them easy to comprehend. Here, the principles involved are conveyed in a variety of algorithms without getting involved in intricate details or tricky implementation issues. To make progress in the application of machine learning techniques to particular data mining problems, it is essential to be able to measure how well you are doing. Chapter 5, which can be read out of sequence, equips the reader to evaluate the results that are obtained from machine learning, addressing the sometimes complex issues involved in performance evaluation.

Part II introduces advanced techniques of data mining. At the lowest and most detailed level, Chapter 6 exposes in naked detail the nitty-gritty issues of implementing a spectrum of machine learning algorithms, including the complexities that are necessary for them to work well in practice (but omitting the heavy mathematical

machinery that is required for a few of the algorithms). Although many readers may want to ignore such detailed information, it is at this level that the full, working, tested Java implementations of machine learning schemes are written. Chapter 7 describes practical topics involved with engineering the input and output to machine learning—for example, selecting and discretizing attributes—while Chapter 8 covers techniques of “ensemble learning,” which combine the output from different learning techniques. Chapter 9 looks to the future.

The book describes most methods used in practical machine learning. However, it does not cover reinforcement learning because that is rarely applied in practical data mining; nor does it cover genetic algorithm approaches, because these are really an optimization technique, or relational learning and inductive logic programming because they are not very commonly used in mainstream data mining applications.

Part III describes the Weka data mining workbench, which provides implementations of almost all of the ideas described in Parts I and II. We have done this in order to clearly separate conceptual material from the practical aspects of how to use Weka. At the end of each chapter in Parts I and II are pointers to related Weka algorithms in Part III. You can ignore these, or look at them as you go along, or skip directly to Part III if you are in a hurry to get on with analyzing your data and don’t want to be bothered with the technical details of how the algorithms work.

Java has been chosen for the implementations of machine learning techniques that accompany this book because, as an object-oriented programming language, it allows a uniform interface to learning schemes and methods for pre- and postprocessing. We chose it over other object-oriented languages because programs written in Java can be run on almost any computer without having to be recompiled, having to go through complicated installation procedures, or—worst of all—having to change the code itself. A Java program is compiled into byte-code that can be executed on any computer equipped with an appropriate interpreter. This interpreter is called the *Java virtual machine*. Java virtual machines—and, for that matter, Java compilers—are freely available for all important platforms.

Of all programming languages that are widely supported, standardized, and extensively documented, Java seems to be the best choice for the purpose of this book. However, executing a Java program is slower than running a corresponding program written in languages like C or C++ because the virtual machine has to translate the byte-code into machine code before it can be executed. This penalty used to be quite severe, but Java implementations have improved enormously over the past two decades, and in our experience it is now less than a factor of two if the virtual machine uses a *just-in-time compiler*. Instead of translating each byte-code individually, a just-in-time compiler translates whole chunks of byte-code into machine code, thereby achieving significant speedup. However, if this is still too slow for your application, there are compilers that translate Java programs directly into machine code, bypassing the byte-code step. Of course, this code cannot be executed on other platforms, thereby sacrificing one of Java’s most important advantages.

UPDATED AND REVISED CONTENT

We finished writing the first edition of this book in 1999, the second edition in early 2005, and now, in 2011, we are just polishing this third edition. How things have changed over the past decade! While the basic core of material remains the same, we have made the most opportunities to both update it and to add new material. As a result the book has close to doubled in size to reflect the changes that have taken place. Of course, there have also been errors to fix, errors that we had accumulated in our publicly available errata file (available through the book's home page at <http://www.cs.waikato.ac.nz/ml/weka/book.html>).

Second Edition

The major change in the second edition of the book was a separate part at the end that included all the material on the Weka machine learning workbench. This allowed the main part of the book to stand alone, independent of the workbench, which we have continued in this third edition. At that time, Weka, a widely used and popular feature of the first edition, had just acquired a radical new look in the form of an interactive graphical user interface—or, rather, three separate interactive interfaces—which made it far easier to use. The primary one is the Explorer interface, which gives access to all of Weka's facilities using menu selection and form filling. The others are the Knowledge Flow interface, which allows you to design configurations for streamed data processing, and the Experimenter interface, with which you set up automated experiments that run selected machine learning algorithms with different parameter settings on a corpus of datasets, collect performance statistics, and perform significance tests on the results. These interfaces lower the bar for becoming a practicing data miner, and the second edition included a full description of how to use them.

It also contained much new material that we briefly mention here. We extended the sections on rule learning and cost-sensitive evaluation. Bowing to popular demand, we added information on neural networks: the perceptron and the closely related Winnow algorithm, and the multilayer perceptron and the backpropagation algorithm. Logistic regression was also included. We described how to implement nonlinear decision boundaries using both the kernel perceptron and radial basis function networks, and also included support vector machines for regression. We incorporated a new section on Bayesian networks, again in response to readers' requests and Weka's new capabilities in this regard, with a description of how to learn classifiers based on these networks and how to implement them efficiently using AD-trees.

The previous five years (1999–2004) had seen great interest in data mining for text, and this was reflected in the introduction of string attributes in Weka, multinomial Bayes for document classification, and text transformations. We also described efficient data structures for searching the instance space: k D-trees and ball trees for finding nearest neighbors efficiently and for accelerating distance-based clustering. We described new attribute selection schemes, such as race search and the use of

support vector machines, and new methods for combining models such as additive regression, additive logistic regression, logistic model trees, and option trees. We also covered recent developments in using unlabeled data to improve classification, including the co-training and co-EM methods.

Third Edition

For this third edition, we thoroughly edited the second edition and brought it up to date, including a great many new methods and algorithms. Our basic philosophy has been to bring the book and the Weka software even closer together. Weka now includes implementations of almost all the ideas described in Parts I and II, and vice versa—pretty well everything currently in Weka is covered in this book. We have also included far more references to the literature: This third edition practically triples the number of references that were in the first edition.

As well as becoming far easier to use, Weka has grown beyond recognition over the last decade, and has matured enormously in its data mining capabilities. It now incorporates an unparalleled range of machine learning algorithms and related techniques. This growth has been partly stimulated by recent developments in the field and partly user-led and demand-driven. This puts us in a position where we know a lot about what actual users of data mining want, and we have capitalized on this experience when deciding what to include in this book.

As noted earlier, this new edition is split into three parts, which has involved a certain amount of reorganization. More important, a lot of new material has been added. Here are a few of the highlights.

Chapter 1 includes a section on web mining, and, under ethics, a discussion of how individuals can often be “reidentified” from supposedly anonymized data. A major addition describes techniques for multi-instance learning, in two new sections: basic methods in Section 4.9 and more advanced algorithms in Section 6.10. Chapter 5 contains new material on interactive cost–benefit analysis. There have been a great number of other additions to Chapter 6: cost-complexity pruning, advanced association-rule algorithms that use extended prefix trees to store a compressed version of the dataset in main memory, kernel ridge regression, stochastic gradient descent, and hierarchical clustering methods. The old chapter Engineering the Input and Output has been split into two: Chapter 7 on data transformations (which mostly concern the input) and Chapter 8 on ensemble learning (the output). To the former we have added information on partial least-squares regression, reservoir sampling, one-class learning, decomposing multiclass classification problems into ensembles of nested dichotomies, and calibrating class probabilities. To the latter we have added new material on randomization versus bagging and rotation forests. New sections on data stream learning and web mining have been added to the last chapter of Part II.

Part III, on the Weka data mining workbench, contains a lot of new information. Weka includes many new filters, machine learning algorithms, and attribute selection algorithms, and many new components such as converters for different file formats and parameter optimization algorithms. Indeed, within each of these categories Weka

contains around 50% more algorithms than in the version described in the second edition of this book. All these are documented here. In response to popular demand we have given substantially more detail about the output of the different classifiers and what it all means. One important change is the inclusion of a brand new Chapter 17 that gives several tutorial exercises for the Weka Explorer interface (some of them quite challenging), which we advise new users to work through to get an idea of what Weka can do.

Acknowledgments

Writing the acknowledgments is always the nicest part! A lot of people have helped us, and we relish this opportunity to thank them. This book has arisen out of the machine learning research project in the Computer Science Department at the University of Waikato, New Zealand. We received generous encouragement and assistance from the academic staff members early on in that project: John Cleary, Sally Jo Cunningham, Matt Humphrey, Lyn Hunt, Bob McQueen, Lloyd Smith, and Tony Smith. Special thanks go to Geoff Holmes, the project leader and source of inspiration, and Bernhard Pfahringer, both of whom also had significant input into many different aspects of the Weka software. All who have worked on the machine learning project here have contributed to our thinking: We would particularly like to mention early students Steve Garner, Stuart Inglis, and Craig Nevill-Manning for helping us to get the project off the ground in the beginning, when success was less certain and things were more difficult.

The Weka system that illustrates the ideas in this book forms a crucial component of it. It was conceived by the authors and designed and implemented principally by Eibe Frank, Mark Hall, Peter Reutemann, and Len Trigg, but many people in the machine learning laboratory at Waikato made significant early contributions. Since the first edition of this book, the Weka team has expanded considerably: So many people have contributed that it is impossible to acknowledge everyone properly. We are grateful to Remco Bouckaert for his Bayes net package and many other contributions, Lin Dong for her implementations of multi-instance learning methods, Dale Fletcher for many database-related aspects, James Foulds for his work on multi-instance filtering, Anna Huang for information bottleneck clustering, Martin Gütlein for his work on feature selection, Kathryn Hempstalk for her one-class classifier, Ashraf Kibriya and Richard Kirkby for contributions far too numerous to list, Niels Landwehr for logistic model trees, Chi-Chung Lau for creating all the icons for the Knowledge Flow interface, Abdelaziz Mahoui for the implementation of K^* , Stefan Mutter for association-rule mining, Malcolm Ware for numerous miscellaneous contributions, Haijian Shi for his implementations of tree learners, Marc Sumner for his work on speeding up logistic model trees, Tony Voyle for least-median-of-squares regression, Yong Wang for Pace regression and the original implementation of $M5'$, and Xin Xu for his multi-instance learning package, *JRip*, logistic regression, and many other contributions. Our sincere thanks go to all these people for their dedicated work, and also to the many contributors to Weka from outside our group at Waikato.

Tucked away as we are in a remote (but very pretty) corner of the southern hemisphere, we greatly appreciate the visitors to our department who play a crucial role in acting as sounding boards and helping us to develop our thinking. We would like to mention in particular Rob Holte, Carl Gutwin, and Russell Beale, each of whom visited us for several months; David Aha, who although he only came for a few days did so at an early and fragile stage of the project and performed a great

service by his enthusiasm and encouragement; and Kai Ming Ting, who worked with us for two years on many of the topics described in Chapter 8 and helped to bring us into the mainstream of machine learning. More recent visitors include Arie Ben-David, Carla Brodley, and Stefan Kramer. We would particularly like to thank Albert Bifet, who gave us detailed feedback on a draft version of the third edition, most of which we have incorporated.

Students at Waikato have played a significant role in the development of the project. Many of them are in the above list of Weka contributors, but they have also contributed in other ways. In the early days, Jamie Littin worked on ripple-down rules and relational learning. Brent Martin explored instance-based learning and nested instance-based representations, Murray Fife slaved over relational learning, and Nadeeka Madapathage investigated the use of functional languages for expressing machine learning algorithms. More recently, Kathryn Hempstalk worked on one-class learning and her research informs part of Section 7.5; likewise, Richard Kirkby's research on data streams informs Section 9.3. Some of the exercises in Chapter 17 were devised by Gabi Schmidberger, Richard Kirkby, and Geoff Holmes. Other graduate students have influenced us in numerous ways, particularly Gordon Paynter, YingYing Wen, and Zane Bray, who have worked with us on text mining, and Quan Sun and Xiaofeng Yu. Colleagues Steve Jones and Malika Mahoui have also made far-reaching contributions to these and other machine learning projects. We have also learned much from our many visiting students from Freiburg, including Nils Weidmann.

Ian Witten would like to acknowledge the formative role of his former students at Calgary, particularly Brent Krawchuk, Dave Maulsby, Thong Phan, and Tanja Mitrovic, all of whom helped him develop his early ideas in machine learning, as did faculty members Bruce MacDonald, Brian Gaines, and David Hill at Calgary, and John Andreae at the University of Canterbury.

Eibe Frank is indebted to his former supervisor at the University of Karlsruhe, Klaus-Peter Huber, who infected him with the fascination of machines that learn. On his travels, Eibe has benefited from interactions with Peter Turney, Joel Martin, and Berry de Bruijn in Canada; Luc de Raedt, Christoph Helma, Kristian Kersting, Stefan Kramer, Ulrich Rückert, and Ashwin Srinivasan in Germany.

Mark Hall thanks his former supervisor Lloyd Smith, now at Missouri State University, who exhibited the patience of Job when his thesis drifted from its original topic into the realms of machine learning. The many and varied people who have been part of, or have visited, the machine learning group at the University of Waikato over the years deserve a special thanks for their valuable insights and stimulating discussions.

Rick Adams and David Bevans of Morgan Kaufmann have worked hard to shape this book, and Marilyn Rash, our project manager, has made the process go very smoothly. We would like to thank the librarians of the Repository of Machine Learning Databases at the University of California, Irvine, whose carefully collected datasets have been invaluable in our research.

Our research has been funded by the New Zealand Foundation for Research, Science, and Technology and the Royal Society of New Zealand Marsden Fund. The Department of Computer Science at the University of Waikato has generously supported us in all sorts of ways, and we owe a particular debt of gratitude to Mark Apperley for his enlightened leadership and warm encouragement. Part of the first edition was written while both authors were visiting the University of Calgary, Canada, and the support of the Computer Science department there is gratefully acknowledged, as well as the positive and helpful attitude of the long-suffering students in the machine learning course, on whom we experimented. Part of the second edition was written at the University of Lethbridge in Southern Alberta on a visit supported by Canada's Informatics Circle of Research Excellence.

Last, and most of all, we are grateful to our families and partners. Pam, Anna, and Nikki were all too well aware of the implications of having an author in the house ("Not again!"), but let Ian go ahead and write the book anyway. Julie was always supportive, even when Eibe had to burn the midnight oil in the machine learning lab, and Immo and Ollig provided exciting diversions. Bernadette too was very supportive, somehow managing to keep the combined noise output of Charlotte, Luke, Zach, and Kyle to a level that allowed Mark to concentrate. Among us, we hail from Canada, England, Germany, Ireland, New Zealand, and Samoa: New Zealand has brought us together and provided an ideal, even idyllic, place to do this work.