

Algorithmic Learning in a Random World

Algorithmic Learning in a Random World

Vladimir Vovk

*University of London
Egham, United Kingdom*

Alexander Gammerman

*University of London
Egham, United Kingdom*

Glenn Shafer

*Rutgers Business School
Newark, NJ, USA*



Springer

Vladimir Vovk
Computer Learning Research
Centre,
Dept. of Computer Science
Royal Holloway
University of London
Egham, Surrey TW2 0EX
UK

EMAIL: vovk@cs.rhul.ac.uk

Alexander Gammerman
Computer Learning Research
Centre,
Dept. of Computer Science
Royal Holloway
University of London
Egham, Surrey TW2 0EX
UK

EMAIL: alex@cs.rhul.ac.uk

Glenn Shafer
Dept. of Accounting and
Information Systems
Rutgers Business School,
Newark and New Brunswick
180 University Avenue
Newark NJ 07102

EMAIL:
gshafer@andromeda.rutgers.edu

Library of Congress Cataloging-in-Publication Data

Vovk, Vladimir.

Algorithmic Learning in a Random World / by Vladimir Vovk, Alexander Gammerman, and Glenn Shafer.

p.cm.

Includes Bibliographical references and index.

ISBN 0-387-00152-2 (HC)

ISBN-13: 978-0387-00152-4 (HC)

e-ISBN 0-387-25061-1 (eBK)

e-ISBN-13: 978-038-725061-8 (eBK)

Printed on acid-free paper.

1. Prediction theory. 2. Algorithms. 3. Stochastic processes. I. Gammerman, A. (Alexander) II. Shafer, Glenn, 1946- III. Title.

QA279.2.V68 2005

519.2'87—dc22

2005042556

© 2005 Springer Science+Business Media, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America. (BS/DH)

9 8 7 6 5 4 3 2 1

SPIN 10900581 (HC) / 11399810 (eBK)

springeronline.com

Contents

Preface	XIII
List of principal results	XV
1 Introduction	1
1.1 Machine learning	1
Learning under randomness	2
Learning under unconstrained randomness	3
1.2 A shortcoming of the existing theory	3
The hold-out estimate of confidence	4
The contribution of this book	5
1.3 The on-line transductive framework	5
On-line learning	5
Transduction	6
On-line/off-line and transduction/induction compromises ..	7
1.4 Conformal prediction	7
Nested prediction sets	8
Validity	9
Efficiency	9
Conditionality	11
Flexibility of conformal predictors	11
1.5 Probabilistic prediction under unconstrained randomness	12
Universally consistent probabilistic predictor	12
Probabilistic prediction using a finite data set	12
Venn prediction	13
1.6 Beyond randomness	13
Testing randomness	13
Low-dimensional dynamic models	14
Islands of randomness	14
On-line compression models	15
1.7 Bibliographical remarks	15

2	Conformal prediction	17
2.1	Confidence predictors	17
	Assumptions	17
	Simple predictors and confidence predictors	18
	Validity	19
	Randomized confidence predictors	22
2.2	Conformal predictors	23
	Bags	23
	Nonconformity and conformity	23
	p-values	25
	Definition of conformal predictors	25
	Validity	26
	Smoothed conformal predictors	27
	A general scheme for defining nonconformity	28
2.3	Ridge regression confidence machine	29
	Least squares and ridge regression	29
	Basic RRCM	30
	Two modifications	34
	Dual form ridge regression	35
	Nearest neighbors regression	38
	Experimental results	39
2.4	Are there other ways to achieve validity?	42
2.5	Conformal transducers	44
	Normalized confidence predictors and confidence transducers	46
2.6	Proofs	47
	Proof of Theorem 2.1	47
	Proof of Theorem 2.6	48
	Proof of Proposition 2.7	48
2.7	Bibliographical and historical remarks	49
	Conformal prediction	49
	Least squares and ridge regression	50
	Kernel methods	50
3	Classification with conformal predictors	53
3.1	More ways of computing nonconformity scores	54
	Nonconformity scores from nearest neighbors	54
	Nonconformity scores from support vector machines	56
	Reducing classification problems to the binary case	58
3.2	Universal predictor	59
3.3	Construction of a universal predictor	61
	Preliminaries	61
	Conformal prediction in the current context	62
	Universal predictor	63
3.4	Fine details of confidence prediction	64

	Bayes confidence predictor	69
3.5	Proofs	70
	Proof of Proposition 3.2	74
	Proof of Proposition 3.3	77
	Proof sketch of Proposition 3.4	80
	Proof sketch of Proposition 3.5	83
3.6	Bibliographical remarks	95
	Examples of nonconformity measures	95
	Universal predictor	95
	Alternative protocols.....	96
	Confidence and credibility	96
4	Modifications of conformal predictors	97
4.1	Inductive conformal predictors	98
	The general scheme for defining nonconformity	99
4.2	Further ways of computing nonconformity scores	100
	De-Bayesing	102
	Bootstrap	103
	Decision trees	104
	Boosting	104
	Neural networks	105
	Logistic regression	106
4.3	Weak teachers	106
	Imperfectly taught predictors	107
	Weak validity	108
	Strong validity	109
	Iterated logarithm validity	109
	Efficiency.....	109
4.4	Off-line conformal predictors and semi-off-line ICPs	110
4.5	Mondrian conformal predictors	114
	Mondrian conformal transducers.....	114
	Using Mondrian conformal transducers for prediction	115
	Generality of Mondrian taxonomies	116
	Conformal transducers	117
	Inductive conformal transducers	118
	Label-conditional Mondrian conformal transducers.....	120
	Attribute-conditional Mondrian conformal transducers ...	120
	Slow teacher	122
4.6	Proofs	123
	Proof of Theorem 4.2, I: $n_k/n_{k-1} \rightarrow 1$ is sufficient	123
	Proof of Theorem 4.2, II: $n_k/n_{k-1} \rightarrow 1$ is necessary	126
	Proof of Theorem 4.4	127
	Proof of Theorem 4.8	128
4.7	Bibliographical remarks	129
	Computationally efficient hedged prediction.....	129

	Specific learning algorithms and nonconformity measures .	130
	Weak teachers	130
	Mondrian conformal predictors	130
5	Probabilistic prediction I: impossibility results	131
5.1	Diverse data sets	132
5.2	Impossibility of estimation of probabilities	132
	Binary case	133
	Multi-label case	134
5.3	Proof of Theorem 5.2	135
	Probability estimators and statistical tests	135
	Complete statistical tests	136
	Restatement of the theorem in terms of statistical tests ...	136
	The proof	138
5.4	Bibliographical remarks and addenda	138
	Density estimation, regression estimation, and regression with deterministic objects	138
	Universal probabilistic predictors	139
	Algorithmic randomness perspective	140
6	Probabilistic prediction II: Venn predictors	143
6.1	On-line probabilistic prediction	145
	The on-line protocol for probabilistic prediction	146
	An informal look at testing calibration	147
	Testing using events of small probability	148
	Calibration events	150
	Testing using nonnegative supermartingales	150
	Predictor has no satisfactory strategy	154
6.2	On-line multiprobability prediction	156
	The on-line protocol	157
	Validity	158
6.3	Venn predictors	158
	The problem of the reference class	159
	Empirical results	161
	Probabilities vs. p-values	162
6.4	A universal Venn predictor	163
6.5	Proofs	163
	Proof of Theorem 6.5	163
	Equivalence of the two definitions of upper probability ...	164
	Proof of Theorem 6.7	166
6.6	Bibliographical remarks	168
	Testing	168
	Frequentist probability	168

7	Beyond exchangeability	169
7.1	Testing exchangeability	169
	Exchangeability supermartingales	170
	Power supermartingales and the simple mixture	171
	Tracking the best power martingale	173
7.2	Low-dimensional dynamic models	177
7.3	Islands of randomness	182
	A sufficient condition for asymptotic validity	183
	Markov sequences	184
7.4	Proof of Proposition 7.2	185
7.5	Bibliographical remarks	187
8	On-line compression modeling I: conformal prediction	189
8.1	On-line compression models	190
8.2	Conformal transducers and validity of OCM	192
	Finite-horizon result	193
8.3	Repetitive structures	195
8.4	Exchangeability model and its modifications	196
	Exchangeability model	196
	Generality and specificity	197
	Inductive-exchangeability models	197
	Mondrian-exchangeability models	198
8.5	Gaussian model	199
	Gauss linear model	201
	Student predictor vs. ridge regression confidence machine	203
8.6	Markov model	205
8.7	Proof of Theorem 8.2	211
8.8	Bibliographical remarks and addenda	214
	Kolmogorov's program	214
	Repetitive structures	215
	Exchangeability model	217
	Gaussian model	217
	Markov model	217
	Kolmogorov's modeling vs. standard statistical modeling	218
9	On-line compression modeling II: Venn prediction	223
9.1	Venn prediction in on-line compression models	224
9.2	Generality of finitary repetitive structures	224
9.3	Hypergraphical models	225
	Hypergraphical repetitive structures	226
	Fully conditional Venn predictor	227
	Generality of hypergraphical models	227
9.4	Junction-tree models	228
	Combinatorics of junction-tree models	229
	Shuffling data sets	230

Decomposability in junction-tree models	231
Prediction in junction-tree models	231
Universality of the fully conditional Venn predictor	233
9.5 Causal networks and a simple experiment	233
9.6 Proofs and further information	236
Proof of Theorem 9.1	236
Maximum-likelihood estimation in junction-tree models	238
9.7 Bibliographical remarks	240
Additive models	240
10 Perspectives and contrasts	241
10.1 Inductive learning	242
Jacob Bernoulli's learning problem	243
Statistical learning theory	246
The quest for data-dependent bounds	249
The hold-out estimate	250
On-line inductive learning	251
10.2 Transductive learning	253
Student and Fisher	254
Tolerance regions	256
Transduction in statistical learning theory	258
PAC transduction	260
Why on-line transduction makes sense	262
10.3 Bayesian learning	264
Bayesian ridge regression	264
Experimental results	267
10.4 Proofs	270
Proof of Proposition 10.1	270
Proof of Proposition 10.2	271
10.5 Bibliographical remarks	272
Inductive prediction	272
Transductive prediction	273
Bayesian prediction	273
Appendix A: Probability theory	275
A.1 Basics	275
Kolmogorov's axioms	275
Convergence	277
A.2 Independence and products	277
Products of probability spaces	277
Randomness model	278
A.3 Expectations and conditional expectations	279
A.4 Markov kernels and regular conditional distributions	280
Regular conditional distributions	281
A.5 Exchangeability	282

Conditional probabilities given a bag	283
A.6 Theory of martingales	284
Limit theorems	286
A.7 Hoeffding's inequality and McDiarmid's theorem	287
A.8 Bibliographical remarks	289
Conditional probabilities	289
Martingales	290
Hoeffding's inequality and McDiarmid's theorem	290
Appendix B: Data sets	291
B.1 USPS data set	291
B.2 Boston Housing data set	292
B.3 Normalization	292
B.4 Randomization and reshuffling	293
B.5 Bibliographical remarks	294
Appendix C: FAQ	295
C.1 Unusual features of conformal prediction	295
C.2 Conformal prediction vs. standard methods	296
Notation	299
References	303
Index	317

Preface

This book is about prediction algorithms that learn. The predictions these algorithms make are often imperfect, but they improve over time, and they are *hedged*: they incorporate a valid indication of their own accuracy and reliability. In most of the book we make the standard assumption of randomness: the examples the algorithm sees are drawn from some probability distribution, independently of one another. The main novelty of the book is that our algorithms learn and predict simultaneously, continually improving their performance as they make each new prediction and find out how accurate it is. It might seem surprising that this should be novel, but most existing algorithms for hedged prediction first learn from a training data set and then predict without ever learning again. The few algorithms that do learn and predict simultaneously do not produce hedged predictions. In later chapters we relax the assumption of randomness to the assumption that the data come from an on-line compression model. We have written the book for researchers in and users of the theory of prediction under randomness, but it may also be useful to those in other disciplines who are interested in prediction under uncertainty.

This book has its roots in a series of discussions at Royal Holloway, University of London, in the summer of 1996, involving AG, Vladimir N. Vapnik and VV. Vapnik, who was then based at AT&T Laboratories in New Jersey, was visiting the Department of Computer Science at Royal Holloway for a couple of months as a part-time professor. VV had just joined the department, after a year at the Center for Advanced Study in Behavioral Sciences at Stanford. AG had become the head of department in 1995 and invited both Vapnik and VV to join the department as part of his program (enthusiastically supported by Norman Gower, the college principal) of creating a machine learning center at Royal Holloway. The discussions were mainly concerned with Vapnik's work on support vector machines, and it was then that it was realized that the number of support vectors used by such a machine could serve as a basis for hedged prediction.

Our subsequent work on this idea involved several doctoral students at Royal Holloway. Ilia Nourtdinov has made several valuable theoretical contributions. Our other students working on this topic included Craig Saunders, Tom Melliush, Kostas Proedrou, Harris Papadopoulos, David Surkov, Leo Gordon, Tony Bellotti, Daniil Ryabko, and David Lindsay. The contribution of Yura Kalnishkan and Misha Vyugin to this book was less direct, mainly through their work on predictive complexity, but still important. Thank you all.

GS joined the project only in the autumn of 2003, although he had earlier helped develop some of its key ideas through his work with VV on game-theoretic probability; see their joint book – *Probability and Finance: It's Only a Game!* – published in 2001.

Steffen Lauritzen introduced both GS and VV to repetitive structures. In VV's case, the occasion was a pleasant symposium organized and hosted by Lauritzen in Aalborg in June 1994. We have also had helpful conversations with Masafumi Akahira, Satoshi Aoki, Peter Bramley, John Campbell, Alexey Chervonenkis, Philip Dawid, José Gonzáles, Thore Graepel, Gregory Gutin, David Hand, Fumiyasu Komaki, Leonid Levin, Xiao Hui Liu, George Loizou, Zhiyuan Luo, Per Martin-Löf, Sally McClean, Boris Mirkin, Fionn Murtagh, John Shawe-Taylor, Sasha Shen', Akimichi Takemura, Kei Takeuchi, Roger Thatcher, Vladimir V'yugin, David Waltz, and Chris Watkins.

Many ideas in this book have their origin in our attempts to understand the mathematical and philosophical legacy of Andrei Nikolaevich Kolmogorov. Kolmogorov's algorithmic notions of complexity and randomness (especially as developed by Martin-Löf and Levin) have been for us the main source of intuition, although they almost disappeared in the final version. VV is grateful to Andrei Nikolaevich for steering him in the direction of compression modeling and for his insistence on its independent value.

We thank the following bodies for generous financial support: EPSRC through grants GR/L35812, GR/M14937, GR/M16856, and GR/R46670; BBSRC through grant 111/BIO14428; MRC through grant S505/65; the Royal Society; the European Commission through grant IST-1999-10226; NSF through grant 5-26830.

University of London (VV, AG, and GS)
Rutgers University (GS)
July 2004

Vladimir Vovk
Alexander Gammernan
Glenn Shafer

List of principal results

<i>Theorem</i>	<i>Topic and page</i>
8.1	Conformal predictors are valid, 193
9.1	Venn predictors are valid, 224
2.6	Conformal predictors are the only valid confidence predictors in a natural class, 43
3.1	There exists an asymptotically optimal confidence predictor (an explicitly constructed conformal predictor), 61
6.7	There exists an asymptotically efficient Venn predictor (explicitly constructed in the proof), 163
4.2	Characterization of teaching schedules under which smoothed conformal predictors are asymptotically valid in probability, 108
4.4	Asymptotic validity of smoothed conformal predictors taught by weak teachers, 109
4.8	Asymptotic efficiency of smoothed conformal predictors taught by weak teachers, 110
5.2	Impossibility of probability estimation from diverse data sets under unconstrained randomness, 134
2.1	There are no exactly valid confidence predictors, 21
6.5	No probabilistic predictor is well calibrated under unconstrained randomness, 155

Introduction

In this introductory chapter, we sketch the existing work in machine learning on which we build and then outline the contents of the book.

1.1 Machine learning

The rapid development of computer technology during the last several decades has made it possible to solve ever more difficult problems in a wide variety of fields. The development of software has been essential to this progress. The painstaking programming in machine code or assembly language that was once required to solve even simple problems has been replaced by programming in high-level object-oriented languages. We are concerned with the next natural step is this progression – the development of programs that can *learn*, i.e., automatically improve their performance with experience.

The need for programs that can learn was already recognized by Alan Turing (1950), who argued that it may be too ambitious to write from scratch programs for tasks that even humans must learn to perform. Consider, for example, the problem of recognizing hand-written digits. We are not born able to perform this task, but we learn to do it quite robustly. Even when the hand-written digit is represented as a gray-scale matrix, as in Fig. 1.1, we can recognize it easily, and our ability to do so scarcely diminishes when it is slightly rotated or otherwise perturbed. We do not know how to write instructions for a computer that will produce equally robust performance.

The essential difference between a program that implements instructions for a particular task and a program that learns is adaptability. A single learn-



Fig. 1.1. A hand-written digit

ing program may be able to learn a wide variety of tasks: recognizing hand-written digits and faces, diagnosing patients in a hospital, estimating house prices, etc.

Recognition, diagnosis, and estimation can all be thought of as special cases of prediction. A person or a computer is given certain information and asked to predict the answer to a question. A broad discussion of learning would go beyond prediction to consider the problems faced by a robot, who needs to act as well as predict. The literature on machine learning, has emphasized prediction, however, and the research reported in this book is in that tradition. We are interested in algorithms that learn to predict well.

Learning under randomness

One learns from experience. This is as true for a computer as it is for a human being. In order for there to be something to learn there must be some stability in the environment; it must be governed by constant, or evolving only slowly, laws. And when we learn to predict well, we may claim to have learned something about that environment.

The traditional way of making the idea of a stable environment precise is to assume that it generates a sequence of examples randomly from some fixed probability distribution, say Q , on a fixed space of possible examples, say \mathbf{Z} . These mathematical objects, \mathbf{Z} and Q , describe the environment.

The environment can be very complex; \mathbf{Z} can be large and structured in a complex way. This is illustrated by the USPS data set from which Fig. 1.1 is drawn (see Appendix B). Here an example is any 16×16 image with 31 shades of gray, together with the digit the image represents (an integer between 0 to 9). So there are $31^{16 \times 16} \times 10$ (this is approximately 10^{382}) possible examples in the space \mathbf{Z} .

In most of this book, we assume that each example consists of an *object* and its *label*. In the USPS dataset, for example, an object is a gray-scale matrix like the one in Fig. 1.1, and its label is the integer between 0 and 9 represented by the gray-scale matrix.

In the problem of recognizing hand-written digits and other typical machine-learning problems, it is the space of objects, the space of possible gray-scale images, that is large. The space of labels is either a small finite set (in what is called *classification problems*) or the set of real numbers (*regression problems*).

When we say that the examples are chosen randomly from Q , we mean that they are independent in the sense of probability theory and all have the distribution Q . They are independent and identically distributed. We call this the *randomness assumption*.

Of course, not all work in machine learning is concerned with learning under randomness. In learning with expert advice, for example, randomness is replaced by a game-theoretic set-up (Vovk 2001a); here a typical result is that the learner can predict almost as well as the best strategy in a pool of

possible strategies. In reinforcement learning, which is concerned with rational decision-making in a dynamic environment (Sutton and Barto 1998), the standard assumption is Markovian. In this book, we will consider extensions of learning under randomness in Chaps. 7–9.

Learning under unconstrained randomness

Sometimes we make the randomness assumption without assuming anything more about the environment: we know the space of examples \mathbf{Z} , we know that examples are drawn independently from the same distribution, and this is all we know. We know nothing at the outset about the probability distribution Q from which each example is drawn. In this case, we say we are *learning under unconstrained randomness*. Most of the work in this book, like much other work in machine learning, is concerned with learning under unconstrained randomness.

The strength of modern machine-learning methods often lies in their ability to make hedged predictions under unconstrained randomness in a *high-dimensional* environment, where examples have a very large (or infinite) number of components. We already mentioned the USPS data set, where each example consists of 257 components (16×16 pixels and the label). In machine learning, this number is now considered small, and the problem of learning from the USPS dataset is sometimes regarded as a toy problem.

1.2 A shortcoming of the existing theory

Machine learning has made significant strides in its study of learning under unconstrained randomness. We now have a wide range of algorithms that often work very well in practice: decision trees, neural networks, nearest neighbors algorithms, and naive Bayes methods have been used for decades; newer algorithms include support vector machines and boosting, an algorithm that is used to improve the quality of other algorithms.

From a theoretical point of view, machine learning’s most significant contributions to learning under unconstrained randomness are comprised by *statistical learning theory*. This theory, which began with the discovery of VC dimension by Vapnik and Chervonenkis in the late 1960s and was partially rediscovered independently by Valiant (1984), has produced both deep mathematical results and learning algorithms that work very well in practice (see Vapnik 1998 for a recent review).

Given a “training” set of examples, statistical learning theory produces what we call a *prediction rule* – a function mapping the objects into the labels. Formally, the value taken by a prediction rule on a new object is a *simple prediction* – a guess that is not accompanied by any statement concerning how accurate it is likely to be. The theory does guarantee, however, that as the training set becomes bigger and bigger these predictions will become more

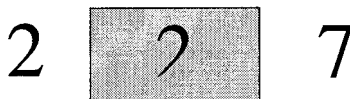


Fig. 1.2. In the problem of digit recognition, we would like to attach lower confidence to the prediction for the image in the middle than to the predictions for the images on the left and the right

and more accurate with greater and greater probability: they are *probably approximately correct*.

How probably and how approximately? This question has not been answered as well as we might like. This is because the theoretical results that might be thought to answer it, the bounds that demonstrate arbitrarily good accuracy with sufficiently large sizes of the training set, are usually too loose to tell us anything interesting for training sets that we actually have. This happens in spite of the empirical fact that the predictions often perform very well in practice. Consider, for example, the problem of recognizing hand-written digits, which we have already discussed. Here we are interested in giving an upper bound on the probability that our learning algorithm fails to choose the right digit; we might like this probability to be less than 0.05, for example, so that we can be 95% confident that the prediction is correct. Unfortunately, typical upper bounds on the probability of error provided by the theory, even for relatively clean data sets such as the USPS data set we have discussed, are greater than 1; bounds less than 1 can usually be achieved only for very straightforward problems or with very large data sets. This is true even for newer results in which the bound on the accuracy depends on the training set (as in, e.g., Littlestone and Warmuth 1986, Floyd and Warmuth 1995; cf. §10.1).

The hold-out estimate of confidence

Fortunately, there are less theoretical and more effective ways of estimating the confidence we should have in predictions output by machine-learning algorithms, including those output by the algorithms proposed by statistical learning theory. One of the most effective is the oldest and most naive: the “hold-out” estimate. In order to compute this estimate, we split the available examples into two parts, a training set and a “test” set. We apply the algorithm to the training set in order to find a prediction rule, and then we apply this prediction rule to the test set. The observed rate of errors on the test set tells us how confident we should be in the prediction rule when we apply it to new examples (for details, see §10.1).

The contribution of this book

When we use a hold-out sample to obtain a meaningful bound on the probability of error, or when we use an error bound from statistical learning theory, we are *hedging* the prediction – we are adding to it a statement about how strongly we believe it. In this book, we develop a different way of producing hedged predictions. Aside from the elegance of our new methods, at least in comparison with the procedure that relies on a hold-out sample, the methods we develop have several important advantages.

As already mentioned in the preface, we do not have the rigid separation between learning and prediction, which is the feature of the traditional approaches that makes hedged prediction feasible. In our basic learning protocol learning and prediction are blended, yet our predictions are hedged.

Second, the hedged predictions produced by our new algorithms are much more confident and accurate. We have, of course, a different notion of a hedged prediction, so the comparison can be only informal; but the difference is so big that there is little doubt that the improvement is real from the practical point of view.

A third advantage of our methods is that the confidence with which the label of a new object is predicted is always tailored not only to the previously seen examples but also to that object.

1.3 The on-line transductive framework

The new methods presented in this book are quite general; they can be tried out, at least, in almost any problem of learning under randomness. The framework in which we introduce and study these methods is somewhat unusual, however. Most previous theoretical work in machine learning has been in an *inductive* and *off-line* framework: one uses a batch of old examples to produce a prediction rule, which is then applied to new examples. We begin instead with a framework that is *transductive*, in the sense advocated by Vapnik (1995, 1998), and *on-line*: one makes predictions sequentially, basing each new prediction on all the previous examples instead of repeatedly using a rule constructed from a fixed batch of examples.

On-line learning

Our framework is *on-line* because we assume that the examples are presented one by one. Each time, we observe the object and predict the label. Then we observe the label and go on to the next example. We start by observing the first object x_1 and predicting its label y_1 . Then we observe y_1 and the second object x_2 , and predict its label y_2 . And so on. At the n th step, we have observed the previous examples

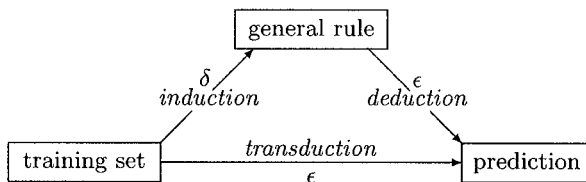


Fig. 1.3. Inductive and transductive prediction

$$(x_1, y_1), \dots, (x_{n-1}, y_{n-1})$$

and the new object x_n , and our task is to predict y_n . The quality of our predictions should improve as we accumulate more and more old examples. This is the sense in which we are learning.

Transduction

Vapnik's distinction between induction and transduction, as applied to the problem of prediction, is depicted in Fig. 1.3. In *inductive prediction* we first move from examples in hand to some more or less general rule, which we might call a prediction or decision rule, a model, or a theory; this is the *inductive step*. When presented with a new object, we derive a prediction from the general rule; this is the *deductive step*. In *transductive prediction*, we take a shortcut, moving from the old examples directly to the prediction about the new object.

Typical examples of the inductive step are estimating parameters in statistics and finding a “concept” (to use Valiant’s 1984 terminology) in statistical learning theory. Examples of transductive prediction are estimation of future observations in statistics (see, e.g., Cox and Hinkley 1974, §7.5) and nearest neighbors algorithms in machine learning.

In the case of simple predictions the distinction between induction and transduction is less than crisp. A method for doing transduction, in our on-line setting, is a method for predicting y_n from $x_1, y_1, \dots, x_{n-1}, y_{n-1}, x_n$. Such a method gives a prediction for any object that might be presented as x_n , and so it defines, at least implicitly, a rule, which might be extracted from $x_1, y_1, \dots, x_{n-1}, y_{n-1}$ (induction), stored, and then subsequently applied to x_n to predict y_n (deduction). So any real distinction is really at a practical and computational level: do we extract and store the general rule or not?

For hedged predictions the difference between transduction and induction goes deeper. We will typically want different notions of hedged prediction in the two frameworks. Mathematical results about induction typically involve two parameters, often denoted ϵ (the desired accuracy of the prediction rule) and δ (the probability of achieving the accuracy of ϵ), whereas results about transduction involve only one parameter, which we will denote ϵ (in this book,

the probability of error we are willing to tolerate); see Fig. 1.3. A detailed discussion can be found in Chap. 10, which also contains a historical perspective on the three main approaches to hedged prediction (inductive, Bayesian, and transductive).

On-line/off-line and transduction/induction compromises

When we work on-line, we would want to use a general rule extracted from $x_1, y_1, \dots, x_{n-1}, y_{n-1}$ only once, to predict y_n from x_n . After observing x_n and then y_n , we have a larger dataset, $x_1, y_1, \dots, x_n, y_n$, and we can use it to extract a new, possibly improved, general rule before trying to predict y_{n+1} from x_{n+1} . So from a purely conceptual point of view, induction seems silly in the on-line framework; it is more natural to say that we are doing transduction, even in cases where the general rule is easy to extract. As a practical matter, however, the computational cost of a transductive method may be high, and in this case, it may be sensible to compromise with the off-line or inductive approach. After accumulating a certain number of examples, we might extract a general rule and use it for a while, only updating it as frequently as is practical.

The methods we present in this book are most naturally described and are most amenable to mathematical analysis in the on-line framework. So we work out our basic theory in that framework, and this theory can be considered transductive. The theory extends, however, to the transductive/inductive compromise just described, where a general rule is extracted and used for a period of time before it is updated (see §4.1).

The theory also extends to relaxations of the on-line protocol that make it close to the off-line setting, and this is important, because most practical problems have at least some off-line aspects. If we are concerned with recognizing hand-written zip codes, for example, we cannot always rely on a human teacher to tell us the correct interpretation of each hand-written zip code; why not use such an *ideal teacher* directly for prediction? The relaxation of the on-line protocol considered in §4.3 includes “slow teachers”, who provide the feedback with a delay, and “lazy teachers”, who provide feedback only occasionally. In the example of zip codes recognition, this relaxation allows us to replace constant supervision by using returned letters for teaching or by occasional lessons.

1.4 Conformal prediction

Most of this book is devoted to a particular method that we call “conformal prediction”. When we use this method, we predict that a new object will have a label that makes it similar to the old examples in some specified way, and we use the degree to which the specified type of similarity holds within

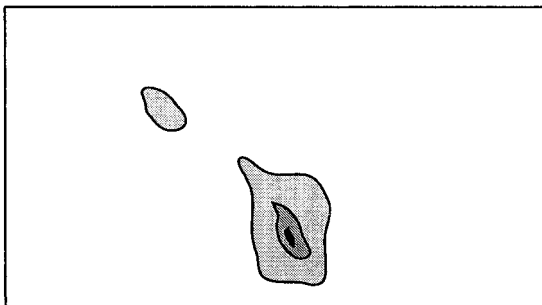


Fig. 1.4. An example of a nested family of prediction sets (casual prediction in black, confident prediction in dark gray, and highly confident prediction in light gray)

the old examples to estimate our confidence in the prediction. Our conformal predictors are, in other words, “confidence predictors”.

We need not explain here exactly how conformal prediction works. This is the topic of the next chapter. But we will explain informally what a confidence predictor aims to do and what it means for it to be valid and efficient.

Nested prediction sets

Suppose we want to pinpoint a target that lies somewhere within a rectangular field. This could be an on-line prediction problem; for each example, we predict the coordinates $y_n \in [a_1, a_2] \times [b_1, b_2]$ of the target from a set of measurements x_n .

We can hardly hope to predict the coordinates y_n exactly. But we can hope to have a method that gives a subset Γ_n of $[a_1, a_2] \times [b_1, b_2]$ where we can be confident y_n lies. Intuitively, the size of the *prediction set* Γ_n should depend on how great a probability of error we want to allow, and in order to get a clear picture, we should specify several such probabilities. We might, for example, specify the probabilities 1%, 5%, and 20%, corresponding to *confidence levels* 99%, 95%, and 80%. When the probability of the prediction set failing to include y_n is only 1%, we declare 99% confidence in the set (highly confident prediction). When it is 5%, we declare 95% confidence (confident prediction). When it is 20%, we declare 80% confidence (casual prediction). We might also want a 100% confidence set, but in practice this might be the whole field assumed at the outset to contain the target.

Figure 1.4 shows how such a family of prediction sets might look. The casual prediction pinpoints the target quite well, but we know that this kind of prediction can be wrong 20% of the time. The confident prediction is much bigger. If we want to be highly confident (make a mistake only for each 100th example, on average), we must accept an even lower accuracy; there is even a completely different location that we cannot rule out at this level of confidence.

In principle, a confidence predictor outputs prediction sets for all confidence levels, and these sets are nested, as in Fig. 1.4.

There are two important desiderata for a confidence predictor:

- They should be *valid*, in the sense that in the long run the frequency¹ of error does not exceed ϵ at each chosen confidence level $1 - \epsilon$.
- They should be *efficient*, in the sense that the prediction sets they output are as small as possible.

We would also like the predictor to be as conditional as possible – we want it to take full account of how difficult the particular example is.

Validity

Our conformal predictors are always valid. Fig. 1.5 shows the empirical confirmation of the validity for one particular conformal predictor that we study in Chap. 3. The solid, dash-dot and dotted lines show the cumulative number of errors for the confidence levels 99%, 95%, and 80%, respectively. As expected, the number of errors made grows linearly, and the slope is approximately 20% for the confidence level 80%, 5% for the confidence level 95%, and 1% for the confidence level 99%.

As we will see in Chap. 2, a precise discussion of the validity of conformal predictors actually requires that we distinguish two kinds of validity: conservative and exact. In general, a conformal predictor is conservatively valid: the probability it makes an error when it outputs a $1 - \epsilon$ set (i.e., a prediction set at a confidence level $1 - \epsilon$) is no greater than ϵ , and there is little dependence between errors it makes when predicting successive examples (at successive *trials*, as we will say). This implies, by the law of large numbers, that the long-run frequency of errors at confidence level $1 - \epsilon$ is about ϵ or less. In practice, the conservativeness is often not very great, especially when n is large, and so we get empirical results like those in Fig. 1.5, where the long-run frequency of errors is very close to ϵ . From a theoretical point of view, however, we must introduce a small element of deliberate randomization into the prediction process in order to get exact validity, where the probability of a $1 - \epsilon$ set being in error is exactly ϵ , errors are made independently at different trials, and the long-run frequency of errors converges to ϵ .

Efficiency

Machine learning has been mainly concerned with two types of problems:

- Classification, where the label space \mathbf{Y} is a small finite set (often binary).
- Regression, where the label space is the real line.

¹By “frequency” we usually mean “relative frequency”.

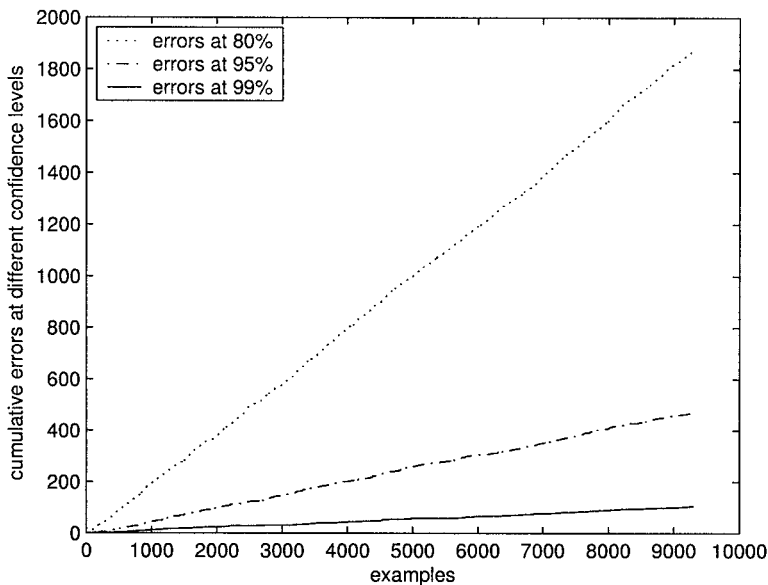


Fig. 1.5. On-line performance of a conformal predictor (“the 1-nearest neighbor conformal predictor”, described in Chap. 3) on the USPS data set (9298 handwritten digits, randomly permuted) for the confidence levels 80%, 95%, and 99%. The figures in this book are not too much affected by statistical variation (due to the random choice of the permutation of the data set)

In classification problems, a natural measure of efficiency of confidence predictors is the number of multiple predictions – the number of prediction sets containing two or more labels, at different confidence levels. In regression problems, the prediction set is often an interval of values, and a natural measure of efficiency of such a prediction is the length of the interval. In §2.3 we use the median length of the convex closures of prediction sets as a measure of efficiency of a sequence of predictions.

As we will see in Chap. 2, there are many conformal predictors for any particular on-line prediction problem, whether it is a classification problem or a regression problem. Indeed, we can construct a conformal predictor from any method for scoring the similarity (conformity, as we call it) of a new example to old ones. All these conformal predictors, it turns out, are valid. Which is most efficient – which produces the smallest prediction sets in practice – will depend on details of the environment that we may not know in advance.

In Chap. 3 we show that there exists a “universal” randomized conformal predictor, making asymptotically no more multiple predictions than any valid confidence predictor. This asymptotic result may, however, have limited relevance to practical prediction problems (as discussed in the next section).

Conditionality

The goal of conditionality can be explained with a simple example discussed by David Cox (1958b). Suppose there are two categories of objects, “easy” (easy to predict) and “hard” (hard to predict). We can tell which objects belong to which category, and the two categories occur with equal probability; about 50% of the objects we encounter are easy, and 50% hard. We have a prediction method that applies to all objects, hard and easy, and has error rate 5%. We do not know what the error rate is for hard objects, but perhaps it is 8%, and we get an overall error rate of 5% only because the rate for easy objects is 2%. In this situation, we may feel uncomfortable, when we encounter a hard object, about appealing to the average error rate of 5% and saying that we are 95% confident of our prediction.

Whenever there are features of objects that we know make the prediction easier or harder, we would like to take these features into account – to condition on them. This is done by conformal predictors almost automatically: they are designed for specific applications so that their predictions take fullest possible account of the individual object to be predicted. What is not achieved automatically is the validity separately for hard and easy objects. It is possible, for example, that if a figure such as Fig. 1.5 were constructed for easy objects only, or for hard objects only, the slopes of the cumulative error lines would be different. We would get the correct slope if we average the slope for easy objects and the slope for hard objects, but we would ideally like to have the “conditional validity”: validity for both categories of objects. As we show in §4.5, this can be achieved by modifying the definition of conformal predictors. In fact, the conditional validity is handled by a general theory that also applies when we segregate examples not by their difficulty but by their time of arrival, as when we are using an inductive rule that we update only at specified intervals.

Flexibility of conformal predictors

A useful feature of our method is that a conformal predictor can be built on top of almost any machine-learning algorithm. The latter, which we call the *underlying algorithm*, may produce hedged predictions, simple predictions, or simple predictions complemented by ad hoc measures of confidence; our experience is that it is always possible to transform it into a conformal predictor that inherits its predictive performance but is, of course, valid, just like any other conformal predictor. In this book we explain how to build conformal predictors using such methods as nearest neighbors, support vector machines, bootstrap, boosting, neural networks, decision trees, ridge regression, logistic regression, and any Bayesian algorithm (see §§2.3, 3.1, 4.2).

1.5 Probabilistic prediction under unconstrained randomness

There are many ways to do classification and regression under unconstrained randomness and for high-dimensional examples. Conformal predictors, for example, combine good theoretical properties with high accuracy in practical problems. It is true that the environment has to be benign, in some sense, for any learning method to be successful, but there are no obvious insurmountable barriers for classification and regression. The situation changes if we move to the harder problem of *probabilistic prediction*: that of guessing the probability distribution for the new object's label. Features of data that can reasonably be expected in typical machine-learning applications become such barriers.

For simplicity, we will assume in this section that the label is binary, 0 or 1. In this case the probabilistic prediction for the label of the new object boils down to one number, the predicted probability that the label is 1.

The problem of probabilistic prediction is discussed in Chaps. 5, 6, and 9. Probabilistic prediction is impossible in an important sense, but there are also senses in which it is possible. So this book gives more than one answer to the question “Is probabilistic prediction possible?” We start with a “yes” answer.

Universally consistent probabilistic predictor

Stone (1977) showed that a nearest neighbors probabilistic predictor (whose probabilistic prediction is the fraction of objects classified as 1 among the k nearest neighbors of the new object, with a suitably chosen k) is *universally consistent*, in the sense that the difference between the probabilistic prediction and the true conditional probability given the object that the label is 1 converges to zero in probability. The only essential assumption is randomness²; there are no restrictive regularity conditions.

Stone's actual result was more general, and it has been further extended in different directions. One of these extensions is used in Chap. 3 for constructing a universal randomized conformal predictor.

Probabilistic prediction using a finite data set

The main obstacle in applying Stone's theorem is that the convergence it asserts is not uniform. The situation that we typically encounter in practice is that we are given a set of examples and a new object and we would like to estimate the probability that the label of the new object is 1. It is well understood that in this situation the applicability of Stone's theorem is very

²The other assumption made by Stone was that the objects were coming from a Euclidean space; since “Euclidean” is equivalent to “Borel” in the context of existence of a universally consistent probabilistic predictor, this assumption is very weak.

limited (see, e.g., Devroye et al. 1996, §7.1). In Chap. 5 we give a new, more direct, formalization of this observation.

We say that a data set consisting of old examples and one new object is *diverse* if no object in it is repeated (in particular, the new object is different from all old objects). The main result of §5.2 asserts that any nontrivial (not empty and not containing 0 and 1) prediction interval for the conditional probability given the new object that the new label is 1 is inadmissible if the data set is diverse and randomness is the only assumption.

The assumption that the data set is diverse is related to the assumption of a high-dimensional environment. If the objects are, for example, complex images, we will not expect precise repetitions among them.

Venn prediction

The results of Chap. 5 show that it is impossible to estimate the true conditional probabilities under the conditions stated; that chapter also contains a result that it is impossible to find conditional probabilities that are as good (in the sense of the algorithmic theory of randomness) as the true probabilities. If, however, we are prepared to settle for less and only want probabilities that are “well calibrated” (in other words, have a frequentist justification), a modification of conformal predictors which we call Venn predictors will achieve this goal, in a very strong non-asymptotic sense. This is the subject of Chap. 6, which is one of the longest in this book. The main problem that we have to deal with in this chapter is that one cannot guarantee that miscalibration will not happen: everything can happen (perhaps with a small probability) for finite sequences and typical probability distributions. But in the case of Venn predictors, any evidence against calibration translates into evidence, at least as strong, against the assumption of randomness; therefore, we expect Venn predictors to be well calibrated as long as we accept the hypothesis of randomness. A significant part of the chapter is devoted to the ways of testing calibration and randomness.

1.6 Beyond randomness

In this book we also consider testing the assumption of randomness and alternatives to this assumption. The most radical alternative is introduced in Chaps. 8 and 9 under the name of “on-line compression modeling”.

Testing randomness

This is the topic of Chap. 7. We start it by adapting the mathematical apparatus developed in the previous chapters to testing the assumption of randomness. The usual statistical approach to testing (sometimes called

the “Neyman–Pearson–Wald” theory) is essentially off-line: in the original Neyman–Pearson approach (see, e.g., Lehmann 1986), the sample size is chosen *a priori*, and in Wald’s (1947) sequential analysis, the sample size is data-dependent but still at some point a categorical decision on whether the null hypothesis is rejected or not is taken (with probability one). The approach of §7.1 is on-line: we constantly update the strength of evidence against the null hypothesis of randomness. Finding evidence against the null hypothesis involves gambling against it, and the strength of evidence equals the gambler’s current capital. For further details and the history of this approach to testing, see Shafer and Vovk 2001. The main mathematical finding of §7.1 is that there exists a wide family of “exchangeability martingales”, which can be successfully applied to detecting lack of randomness.

Low-dimensional dynamic models

The ability to test the assumption of randomness immediately provides opportunities for extending the range of stochastic environments to which one can apply the idea of conformal prediction. In §7.2 we consider the simple case where we are given a parametric family of transformations one of which is believed to transform the observed data sequence into a random sequence. If the parameter is a vector in a low-dimensional linear space, we can often hope to be able to detect lack of randomness of the transformed data sequence for most values of the parameter as the number of observed examples grows. When the range of possible values of the parameter becomes very narrow, conformal prediction can be used.

Islands of randomness

When we are willing to make the assumption of randomness, or some version of this assumption as described in the previous subsection, about a data sequence, it usually means that this data sequence was obtained from a much bigger sequence by careful filtering. When observing the real world around us, we cannot hope that a simple model such as randomness will explain much, but the situation changes if we, e.g., discard all observations except the results of fair coin tosses.

In §7.3, we briefly discuss the case where randomness can appear as a property of only relatively small subsequences of the full data sequence. Such a “big picture” is of great interest to philosophers (see, e.g., Venn 1866). Once we know that some subsequence is random (this knowledge can be based on an initial guess and then using as severe tests as we can think of to try and falsify this guess; §7.1 provides the means for the second stage), we can apply the theory developed under the assumption of randomness to make predictions.

On-line compression models

As we will see in Chap. 8, the idea of conformal prediction generalizes from learning under randomness, where examples are independent and identically distributed, to “on-line compression models”. In an on-line compression model, it is assumed that the data can be summarized in way that can be updated as new examples come in, and the only probabilities given are backward probabilities – probabilities for how the updated summary might have been obtained.

On-line compression models derive from the work of Andrei Kolmogorov. They open a new direction for broadening the applicability of machine-learning methods, giving a new meaning to the familiar idea that learning can be understood as information compression.

In Chap. 8 we consider in detail three important on-line compression models (Gaussian, Markov, exchangeability) and their variants. In Chap. 9 we extend the idea of Venn prediction to on-line compression modeling and apply it to a new model, which we call the “hypergraphical model”.

1.7 Bibliographical remarks

Each chapter of this book ends with a section entitled “Bibliographical remarks”, or similarly. These sections are set in a small font and may use mathematical notions and results not introduced elsewhere in the book.

Turing suggested the idea of machine learning in his paper published in *Mind* as an approach to solving his famous “imitation game” (Turing 1950, §1).

A recent empirical study of various bounds on prediction accuracy is reported in Langford 2004. It found the hold-out estimate to be a top performer.

Mitchell (1997, §8.6) discusses advantages and disadvantages of inductive and transductive approaches to making simple predictions. The near-synonyms for “transductive learning” used in that book are “lazy learning” and “instance-based learning”.

Conformal prediction

In this chapter we formally introduce conformal predictors. After giving the necessary definitions, we will prove that when a conformal predictor is used in the on-line mode, its output is valid, not only in the asymptotic sense that the sets it predicts for any fixed confidence level $1 - \epsilon$ will be wrong with frequency at most ϵ (approaching ϵ in the case of smoothed conformal predictors) in the long run, but also in a much more precise sense: the error probability of a smoothed conformal predictor is ϵ at every trial and errors happen independently at different trials. In §2.4 we will see that conformal prediction is indispensable for achieving this kind of validity. The basic procedure of conformal prediction might look computationally inefficient when the label set is large, but in §2.3 we show that in the case of, e.g., least squares regression (where the label space \mathbb{R} is uncountable) there are ways of making conformal predictors much more efficient.

2.1 Confidence predictors

The conformal predictors we define in this chapter are confidence predictors – they make a range of successively more specific predictions with successively less confidence. In this section we define precisely what we mean by a confidence predictor and its validity.

Assumptions

We assume that Reality outputs successive pairs

$$(x_1, y_1), (x_2, y_2), \dots, \tag{2.1}$$

called *examples*. Each example (x_i, y_i) consists of an *object* x_i and its *label* y_i . The objects are elements of a measurable space \mathbf{X} called the *object space* and the labels are elements of a measurable space \mathbf{Y} called the *label space*.

We assume that \mathbf{X} is non-empty and that \mathbf{Y} contains at least two essentially different elements¹. When we need a more compact notation, we write z_i for (x_i, y_i) . We set

$$\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$$

and call \mathbf{Z} the *example space*. Thus the infinite data sequence (2.1) is an element of the measurable space \mathbf{Z}^∞ .

When we say that the objects are *absent*, we mean that $|\mathbf{X}| = 1$. In this case x_i do not carry any information and do not need to be mentioned; we will then identify \mathbf{Y} and \mathbf{Z} .

Our standard assumption is that Reality chooses the examples independently from some probability distribution Q on \mathbf{Z} – i.e., that the infinite sequence z_1, z_2, \dots is drawn from the *power probability distribution* Q^∞ in \mathbf{Z}^∞ . Most of the results of this book hold under this *randomness assumption*, but usually we need only the slightly weaker assumption that the infinite data sequence (2.1) is drawn from a distribution P on \mathbf{Z}^∞ that is *exchangeable*. The statement that P is exchangeable means that for every positive integer n , every permutation π of $\{1, \dots, n\}$, and every measurable set $E \subseteq \mathbf{Z}^n$,

$$\begin{aligned} P \{ (z_1, z_2, \dots) \in \mathbf{Z}^\infty : (z_1, \dots, z_n) \in E \} \\ = P \{ (z_1, z_2, \dots) \in \mathbf{Z}^\infty : (z_{\pi(1)}, \dots, z_{\pi(n)}) \in E \} . \end{aligned}$$

Every power distribution is exchangeable, and under a natural regularity condition (\mathbf{Z} is a Borel space), any exchangeable distribution on \mathbf{Z}^∞ is a mixture of power distributions; for details, see §A.5. In our mathematical results, we usually use the randomness assumption or the exchangeability assumption depending on which one leads to a stronger statement.

Simple predictors and confidence predictors

We assume that at the n th trial Reality first announces the object x_n and only later announces the label y_n . If we simply want to predict y_n , then we need a function

$$D : \mathbf{Z}^* \times \mathbf{X} \rightarrow \mathbf{Y} . \quad (2.2)$$

We call such a function a *simple predictor*, always assuming it is measurable. For any sequence of old examples, say $x_1, y_1, \dots, x_{n-1}, y_{n-1} \in \mathbf{Z}^*$, and any new object, say $x_n \in \mathbf{X}$, it gives $D(x_1, y_1, \dots, x_{n-1}, y_{n-1}, x_n) \in \mathbf{Y}$ as its prediction for the new label y_n .

As we explained in §1.4, however, we have a more complicated notion of prediction. Instead of merely choosing a single element of \mathbf{Y} as our prediction

¹Formally, the σ -algebra on \mathbf{Y} is assumed to be different from $\{\emptyset, \mathbf{Y}\}$. It is convenient to assume that for each pair of distinct elements of \mathbf{Y} there is a measurable set containing only one of them; we will do this without loss of generality, and then our assumption about \mathbf{Y} is that $|\mathbf{Y}| > 1$.