Department of Computer Science
Series of Publications A
Report A-2007-4

Statistical and Information-Theoretic Methods for Data Analysis

Teemu Roos

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in the auditorium of Arppeanum (Helsinki University Museum, Snellmaninkatu 3) on June 9th, at 12 o'clock noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

Copyright © 2007 Teemu Roos
ISSN 1238-8645
ISBN 978-952-10-3988-1 (paperback)
ISBN 978-952-10-3989-8 (PDF)
Computing Reviews (1998) Classification: G.3, H.1.1, I.2.6, I.2.7, I.4, I.5
Helsinki 2007
Helsinki University Printing House

Statistical and Information-Theoretic Methods for Data Analysis

Teemu Roos

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
teemu.roos@cs.helsinki.fi
http://www.cs.helsinki.fi/teemu.roos/

PhD Thesis, Series of Publications A, Report A-2007-4
Helsinki, March 2007, 82 + 75 pages
ISSN 1238-8645
ISBN 978-952-10-3988-1 (paperback)
ISBN 978-952-10-3989-8 (PDF)

Abstract

In this Thesis, we develop theory and methods for computational data analysis. The problems in data analysis are approached from three perspectives: statistical learning theory, the Bayesian framework, and the information-theoretic minimum description length (MDL) principle. Contributions in statistical learning theory address the possibility of generalization to unseen cases, and regression analysis with partially observed data with an application to mobile device positioning. In the second part of the Thesis, we discuss so called Bayesian network classifiers, and show that they are closely related to logistic regression models. In the final part, we apply the MDL principle to tracing the history of old manuscripts, and to noise reduction in digital signals.

Computing Reviews (1998) Categories and Subject Descriptors:

G.3 Probability and Statistics: correlation and regression analysis, nonparametric statistics
H.1.1 Systems and Information Theory
I.2.6 Learning: concept learning, induction, parameter learning
I.2.7 Natural Language Processing: text analysis
I.4 Image Processing and Computer Vision
I.5 Pattern Recognition

General Terms:
data analysis, statistical modeling, machine learning

Additional Key Words and Phrases:
information theory, statistical learning theory, Bayesianism, minimum description length principle, Bayesian networks, regression, positioning, stemmatology, denoising

Preface

> " We are all shaped by the tools we use, in particular: the formalisms we use shape our thinking habits, for better or for worse [...] "
>
> *Edsger W. Dijkstra (1930–2002)*

This Thesis is about data analysis: learning and making inferences from data. What do the data have to say? To simplify, this is the question we would ultimately like to answer. Here the *data* may be whatever observations we make, be it in the form of labeled feature vectors, text, or images — all of these formats are encountered in this work. Here, as usual, the computer scientist's *modus operandi* is to develop rules and algorithms that can be implemented in a computer. In addition to computer science, there are many other disciplines that are relevant to data analysis, such as statistics, philosophy of science, and various applied sciences, including engineering and bioinformatics. Even these are divided into various subfields. For instance, the Bayesian versus non-Bayesian division related to the interpretation of probability exists in many areas.

Diversity characterizes also the present work. The six publications that make the substance of this Thesis contain only one cross-reference between each other (the fifth paper is cited in the sixth one). The advantage of diversity is that with more tools than just a hammer (or a support vector machine), all problems do not have to be nails. Of course, one could not even hope to be comprehensive and all-inclusive. In all of the following, probability plays a central role, often together with its cousin, the code-length. This defines *ad hoc* the scope and the context of this Thesis. Hence also its title.

In order to cover the necessary preliminaries and background for the actual work, three alternative paradigms for data analysis are encountered before reaching the back cover of this work. The Thesis is divided accordingly into three parts: each part includes a brief introduction to one of the paradigms, followed by contributions in it. These part are: 1. Statistical Learning Theory; 2. the Bayesian Approach; and 3. Minimum Description

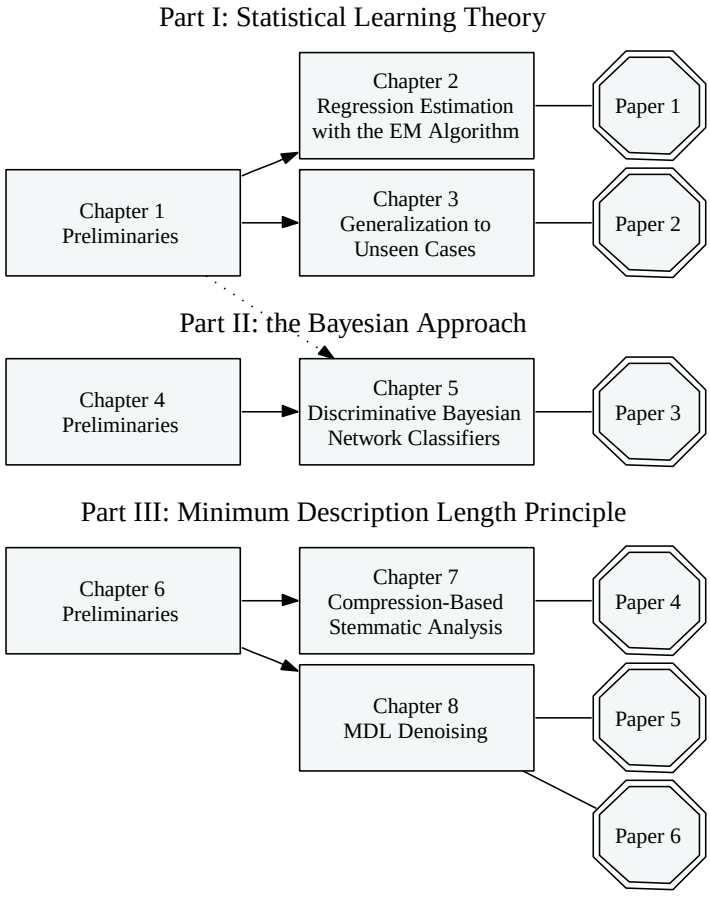

Figure 1: The relationships between the chapters and original publications (Papers 1–6) of the Thesis.

Length Principle. The structure of the Thesis is depicted in Figure 1.

As this is not a textbook intended to be self-contained, many basic concepts are assumed known. Standard references are, for instance, in probability and statistics [28], in machine learning [26, 83], in Bayesian methods [7], and in information theory [19, 37].

Acknowledgments: I am grateful to my advisors, Professors Petri Myllymäki and Henry Tirri, for their advice, for their efforts in managing the CoSCo research group where I have had the pleasure to work for the whole duration of my Ph.D. studies, and for making it possible for me to work with some of the most gifted and acclaimed people in my area. Henry and

Petri showed me that research can, and should, be fun.

The Department of Computer Science of the University of Helsinki, and the Helsinki Institute for Information Technology have provided me with a pleasant working environment in terms of office, computing, and sports facilities. As the project secretary of CoSCo, Mrs. Taina Nikko has saved me from a great deal of paperwork.

In addition to the Department of Computer Science, financial support from the Helsinki Graduate School in Computer Science and Engineering (HECSE), the Academy of Finland, the Finnish Funding Agency for Technology and Innovation (Tekes), the Center for International Mobility (CIMO), the EU Network of Excellence PASCAL, and Tervakosken Opintotukisäätiö is gratefully acknowledged.

I warmly thank all the people that have had — and hopefully continue to have — a significant impact on my work. Among these people two stand out: Professor Emeritus Jorma Rissanen and Dr. Peter Grünwald. Their guidance has been irreplaceable. I also thank Professor Paul Vitányi, Docent Tuomas Heikkilä, and Dr. Wray Buntine. Dr. Mikko Koivisto and Dr. Matti Kääriäinen have always provided educated answers to my questions on machine learning and Bayesian modeling. With my fellow-workers in CoSCo, especially Hannes Wettig and Tomi Silander, I have had countless inspiring discussions on all things related to Bayes, MDL, and what not. I thank all of them for that. The same goes for Dr. Rudi Cilibrasi.

The manuscript of this Thesis was reviewed by Professors Ioan Tabus and Tommi Jaakkola. I thank them for their time and useful comments.

I am grateful to my parents, Antti and Airi, and to my brothers, Pekka and Timo for their support, and for discrete (*and* continuous) inquiries about the progress of my studies. Finally, I dearly thank my beloved wife Eira, and our two sons, Anto and Peik, for their unconditional love. "You are the reason I am, you are all my reasons."

Helsinki, May 16, 2007
Teemu Roos

Original Publications and Contributions

This Thesis is based on the following publications, which are referred to in the text as Papers 1–6.

1. Teemu Roos, Petri Myllymäki, and Henry Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing* 1(1):59–69, 2002.

2. Teemu Roos, Peter Grünwald, Petri Myllymäki, and Henry Tirri. Generalization to unseen cases. In Y. Weiss, B. Schölkopf and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1129–1136. MIT Press, 2006.

3. Teemu Roos, Hannes Wettig, Peter Grünwald, Petri Myllymäki, and Henry Tirri. On discriminative Bayesian network classifiers and logistic regression. *Machine Learning* 59(3):267–296, 2005.

4. Teemu Roos, Tuomas Heikkilä, and Petri Myllymäki. A compression-based method for stemmatic analysis. In G. Brewka, S. Coradeschi, A. Perini and P. Traverso, editors, *Proceedings of the 17th European Conference on Artificial Intelligence*, pages 805-806. IOS Press, 2006.

5. Teemu Roos, Petri Myllymäki, and Henry Tirri. On the behavior of MDL denoising. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 309–316. Society for AI and Statistics, 2005.

6. Teemu Roos, Petri Myllymäki, and Jorma Rissanen. MDL denoising revisited. Submitted for publication, 2006. Preprint available at: arXiv cs.IT/0609138.

The papers are printed in the end of the Thesis. The printed version of Paper 4 is an extended six page version of the two-page summary published in the ECAI Proceedings.

The main contributions of the six papers are:

Paper 1: A regression model is proposed for signal strength readings in mobile devices, and used for estimating the location of the device (positioning). The main technical contribution is an EM algorithm for estimating propagation parameters from partially observable data.

Paper 2: By analyzing classification error on unseen cases, i.e., cases outside the observed training sample, we show for the first time that it is possible to derive distribution-free generalization error bounds for unseen cases. This implies that certain claims attributed to the No Free Lunch theorems are overly pessimistic.

Paper 3: We explicitly formalize the connection between Bayesian network classifiers and logistic regression, and prove equivalence of these two under a graph-theoretic assumption on the Bayesian network structure. Empirical results illustrate some aspects relevant to practical classifier design.

Paper 4: The problem of stemmatology is to reconstruct family trees of texts that are available in several variant readings. We present a compression-based criterion and an algorithm, building upon techniques from bioinformatics and stochastic optimization.

Paper 5: We analyze the performance of an MDL denoising method by Rissanen, and point out a restriction on its range of applicability in both theory and practice. The behavior is explained in terms of a new interpretation of the method.

Paper 6: The new interpretation given in Paper 5 to the earlier MDL method is assumed. This leads to three refinements and extensions, each of which is shown to significantly improve performance in experiments on artificial and real-world signals.

The contributions of the present author are substantial in all papers. The main contributions of Papers 1 & 4–6 are by the present author. In Paper 2, some of the main contributions are due to Dr. Peter Grünwald (including Theorem 2). In Paper 3, some of the main contributions are due to Hannes Wettig (in particular, most of the experimental part) and Dr. Peter Grünwald.

Contents

Part I

Statistical Learning Theory

Chapter 1

Preliminaries

In machine learning, the most commonly assumed framework is that of statistical learning theory (see, for instance, [121, 122, 10] and references therein). It involves an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. The input space contains *instances* x that may be sequences like strings of text, vectors of measurements, or matrices like grayscale bitmap images, etc. Labels y from the output space are attached to the instances. The labels are often nominal or real-valued. The statistical nature of the theory is due to the assumption that independent and identically distributed (i.i.d.) (x, y)-pairs are sampled from a fixed but unknown probability distribution P.

> **A Remark on Mathematical Notation:**[1] Some comments on mathematical notation are in place now, and more will be presented on occasion. Notation is overloaded by using lower-case letters, x, y, θ, etc., to denote both random variables and their values. Domains are denoted by calligraphic letters when available, e.g., $\mathcal{X}, \mathcal{Y}, \Theta$. Letters P, Q, etc. are used to denote probability measures. The corresponding probability mass functions or probability density functions are denoted by the letters p, q, etc. Hence, the often used expression $\Pr[X = x]$, where X is a (discrete) random variable, and x its value, is written here simply as $p(x)$. The expectation of an expression like $\phi(x)$, involving the random variable x, is denoted by $\mathbb{E}_{x\sim P}[\phi(x)]$, where the subscript indicates the variable over which the expectation is taken and the relevant distribution. Whenever the distribution is clear from the context, it is omitted.

A hypothesis is a mapping of the form $h : \mathcal{X} \to \mathcal{D}$, where the decision space $\mathcal{D}$ contains the allowed predictions. A loss function $\ell(y, \tilde{y})$ measures

[1] Remarks and digressions from the main subject are indicated by smaller typeface and indentation, like this paragraph.

the loss incurred by giving prediction $\tilde{y} \in \mathcal{D}$ when the correct label is $y \in \mathcal{Y}$. The *risk* of a given hypothesis h is defined as the expected loss:

$$\mathcal{E}(h) := \mathbb{E}_{(x,y)\sim P}\left[\ell(y, h(x))\right] \ . \tag{1.1}$$

Research in statistical learning theory focuses on topics such as (*i*) constructing learning algorithms that output a hypothesis with small risk when given a training set sampled from the distribution P, and (*ii*) developing guarantees on the performance of such algorithms.

Vapnik lists the following three main problem settings studied in statistical learning theory [122]:

Classification (or Pattern Recognition)**:** The decision space is equal to the output space, often the set $\{\pm 1\}$. Loss is given by the 0/1-loss:

$$\ell_{0/1}(y, \tilde{y}) := \begin{cases} 0 & \text{if } y = \tilde{y}, \\ 1 & \text{otherwise.} \end{cases}$$

The risk of a hypothesis is then the probability of misclassification, also known as *generalization error*. This is minimized by the label y^* with the highest probability: $y^* = \arg\max_y p(y \mid x)$.

Regression Estimation: The decisions and outputs are both real numbers, $\mathcal{D} = \mathcal{Y} = \mathbb{R}$. Loss is given by the squared error:

$$\ell_2(y, \tilde{y}) := (y - \tilde{y})^2 \ .$$

The risk is minimized by $\mathbb{E}_y\left[y \mid x\right]$, the conditional expectation of y.

Density Estimation: Here the outputs are ignored or combined with the inputs to form the pair $z = (x, y)$. The decisions are densities over $\mathcal{X} \times \mathcal{Y}$, and loss is given by the log-loss:

$$\ell_{\ln}(z, \tilde{p}) := -\ln \tilde{p}(z) \ .$$

If the generating distribution is discrete with probability mass function p, the risk is minimized by setting $\tilde{p} = p$, i.e., by using the generating distribution, in which case the risk equals the entropy of p. A similar statement holds for the continuous case as well.

In all three cases it is seen that the optimal decisions depend on the unknown generating distribution P in an essential way. The key point is that the learning algorithm is supposed to work for a large class of generating distributions, or in fact, in the distribution-free setting, for all possible distributions. All information concerning P is extracted from the training set. In many cases this is ultimately based on the law(s) of large numbers applied to relative frequency estimators, as discussed next.

1.1 Generalization error bounds

Let the empirical error of a hypothesis $h : \mathcal{X} \to \mathcal{D}$ be defined as

$$\mathcal{E}^n_{\text{emp}}(h) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i)) \ ,$$

where (x_i, y_i), $1 \leq i \leq n$ are the labeled instances in the training set. Whenever the random variable $\ell(y, h(x))$ has finite mean under distribution P, the empirical error converges in probability[2] to the true risk:

$$\mathcal{E}^n_{\text{emp}}(h) \xrightarrow[n\to\infty]{P} \mathcal{E}(h) \ .$$

In practice, rate of convergence is of great interest. This rate can be characterized by bounds that relate the error of the estimate to sample size. In statistical terms, such bounds are confidence intervals for the true risk.

In the case of classification, where the loss is binary valued, the random variable $n\mathcal{E}^n_{\text{emp}}(h)$ has a binomial distribution with the bias parameter given by the generalization error $\mathcal{E}(h)$. Exact upper (or lower) bounds on $\mathcal{E}(h)$ can be obtained by considering the binomial tail probabilities.

Proposition 1.1 (Binomial tails) *For a fixed probability of error $\mathcal{E}(h)$, the probability of observing more than k errors in n trials is given by*

$$\Pr\left[\mathcal{E}^n_{\text{emp}}(h) > k/n\right] = \sum_{j=k+1}^{n} \binom{n}{j} \mathcal{E}(h)^j (1 - \mathcal{E}(h))^{n-j} \ . \tag{1.2}$$

Having observed $\mathcal{E}^n_{\text{emp}}(h)$, we can find the smallest $\mathcal{E}(h)$ for which the right-hand side is greater than or equal to the required confidence level $1 - \delta$, as illustrated in Fig. 1.1. This gives the smallest possible upper bound: for any value smaller than this, the probability of producing a valid upper bound — larger than or equal to the true value of $\mathcal{E}(h)$ — would be less than $1 - \delta$. This technique is known as binomial tail inversion[3] [62].

There are several lower bounds for the right-hand side of (1.2) that are somewhat easier to use than binomial tail inversion but that either apply only in special cases or that are not exact.

[2] A sequence of random variables $(A_1, A_2, \ldots)$ converges to the scalar a in probability iff for all $\epsilon, \delta > 0$ there exists a number $n_0 = n_0(\epsilon, \delta)$ such that for all $n > n_0$ with probability at least $1 - \delta$ we have $|A_n - a| < \epsilon$.

[3] Programs calculating this and many other bounds are available at `http://hunch.net/~jl/projects/prediction_bounds/prediction_bounds.html`.

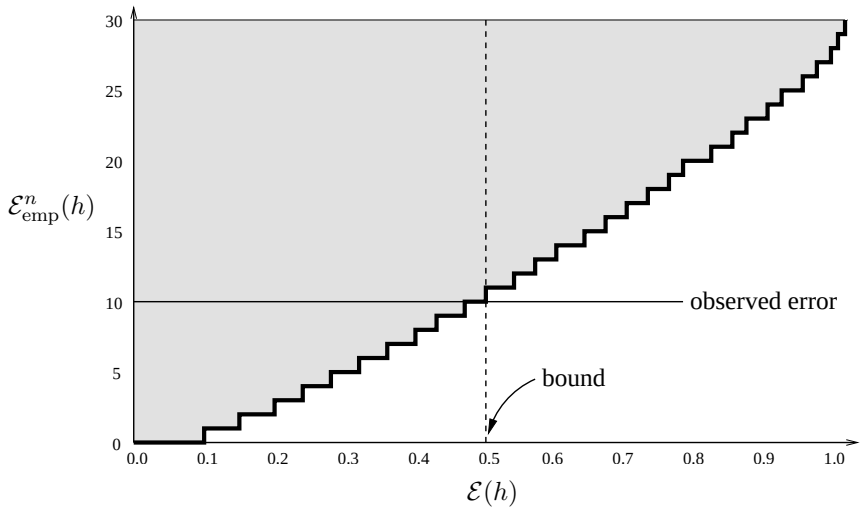

Figure 1.1: Illustration of binomial tail inversion. For each value of $\mathcal{E}(h)$, the shaded area above the bold curve contains at least 95 % of the total probability mass. For $\mathcal{E}^{30}_{\text{emp}}(h) = 10$ the upper bound on $\mathcal{E}(h)$ is 0.499.

Proposition 1.2 (Realizable case) *In the error-free, or* realizable*, case we have*

$$\Pr[\mathcal{E}^n_{\text{emp}}(h) > 0] = 1 - (1 - \mathcal{E}(h))^n \geq 1 - \exp\left(-n\mathcal{E}(h)\right) \quad .$$

Theorem 1.1 (Chernoff bound [15]) *For $k/n < \mathcal{E}(h)$, the probability of observing more than k errors in n trials is lower-bounded by*

$$\Pr\left[\mathcal{E}^n_{\text{emp}}(h) > k/n\right] \geq 1 - \exp\left(-n\text{KL}\left(\frac{k}{n} \,\middle\|\, \mathcal{E}(h)\right)\right) \quad ,$$

where

$$\text{KL}(r \parallel s) := r \ln \frac{r}{s} + (1 - r) \ln \frac{1 - r}{1 - s}$$

is the Kullback-Leibler divergence between two binomial distributions indexed by parameters r and s respectively.

Corollary 1.1 (Hoeffding bound [50]) *For $k/n < \mathcal{E}(h)$, the probability of observing more than k errors in n trials is lower-bounded by*

$$\Pr\left[\mathcal{E}^n_{\text{emp}}(h) > k/n\right] \geq 1 - \exp\left(-2n\left(\mathcal{E}(h) - \frac{k}{n}\right)^2\right) \quad .$$

The corollary follows directly from Thm 1.1 by using the following lower bound on Kullback-Leibler divergence:

$$\mathrm{KL}\left(\frac{k}{n} \,\middle\|\, \mathcal{E}(h)\right) \geq 2\left(\mathcal{E}(h) - \frac{k}{n}\right)^2 \quad .$$

The advantage of Hoeffding's bound compared to the binomial tail bound and the relative entropy Chernoff bound is that it can be easily inverted: we can let $\delta = \exp(-2n(\mathcal{E}(h) - k/n)^2)$ and solve for k/n to find that with probability at least $1 - \delta$ we have

$$\mathcal{E}(h) < \mathcal{E}^n_{\mathrm{emp}}(h) + \sqrt{\frac{\ln(1/\delta)}{2n}} \quad . \tag{1.3}$$

This is really the way we would like the bounds to be expressed since now we have the unknown quantity, $\mathcal{E}(h)$, on one side, and known quantities on the other. Unfortunately, such inverted forms are not available for the binomial tail bound and the relative entropy Chernoff bound. They have to be inverted numerically as described above (Fig. 1.1).

On the other hand, the Hoeffding bound is significantly weaker than either one of the other bounds, especially near the boundaries $k \approx 0$ or $k \approx n$. For instance, consider the realizable case, $\mathcal{E}^n_{\mathrm{emp}}(h) = 0$. It is easily verified that in this case the relative entropy Chernoff bound agrees with the realizable case bound, Prop. 1.2. Inverting the realizable case bound by setting $\delta = \exp(-n\mathcal{E}(h))$ and solving for $\mathcal{E}(h)$ yields

$$\mathcal{E}(h) < \frac{\ln(1/\delta)}{n} \quad . \tag{1.4}$$

This is a significant improvement: the rate $\mathcal{O}(n^{-1/2})$ implied by (1.3) is improved to $\mathcal{O}(n^{-1})$. Unfortunately, the worst-case rate $\mathcal{O}(n^{-1/2})$ that occurs near the error rate $\mathcal{E}^n_{\mathrm{emp}}(h) = 1/2$ cannot be improved upon.

1.2 Complexity regularization

The above bounds apply to a single hypothesis h, but in practice it is often necessary to bound the generalization error for a whole class of hypotheses $\mathcal{H}$ simultaneously. For instance, this is useful for constructing learning algorithms: having bounded the risk of all hypotheses, the bound holds for the particular hypothesis chosen by a learning algorithm. If we were to use the bounds presented above as such for several hypotheses, it would of course still be true that any given bound, singled out in advance, would

hold with high probability. However, if the number of hypotheses is large, it is actually highly unlikely that all the bounds hold at the same time. This is called in statistics the *multiple testing* problem. To avoid it, we have to loosen the bounds by an amount that somehow depends on the complexity of the hypothesis or the hypothesis class. This is known as *complexity regularization*.

The simplest solution, applicable to countable hypothesis classes, is the union bound[4]. Let $\mathcal{H} = \{h_1, h_2, \ldots\}$ be a a countable hypothesis class, and $\{p_1, p_2, \ldots\}$ be a set of numbers that satisfy the formal requirements of a sub-probability distribution, i.e., are non-negative and sum to at most one[5]. Now we can use, for instance, the Hoeffding bound for each of the hypotheses and apply the union bound to obtain the following theorem.

Theorem 1.2 (Occam's razor bound [9, 73]) *With probability at least* $1 - \delta$ *we have*

$$\mathcal{E}(h) < \mathcal{E}^n_{\text{emp}}(h_i) + \sqrt{\frac{\ln(1/p_i) + \ln(1/\delta)}{2n}} \quad \textit{for all } h_i \in \mathcal{H} \ . \tag{1.5}$$

The higher the 'prior' probability p_i of a hypothesis, the tighter the bound. If the class is finite, we can use the uniform distribution which yields $\ln(1/p_i) = \ln|\mathcal{H}|$, where $|\mathcal{H}|$ is the number of hypotheses in the class.

To extend this approach to uncountable hypothesis classes, one can use the fact that even if there are infinitely many hypotheses, the number of different binary predictions on a sample of size n is always at most 2^n. Depending on the hypothesis class, this number may be significantly smaller. The canonical example of this is based on the Vapnik–Chervonenkis (VC) dimension [123]. For classes with finite VC dimension, VCdim, the number of different predictions is upper bounded by $(n+1)^{\text{VCdim}}$, i.e., the number is polynomial instead of exponential in the sample size n.

A more recent approach is based on Rademacher complexity [58, 4].

[4]The union bound (or Boole's inequality) simply states that given a countable set of events with probabilities $(p_1, p_2, \ldots)$, the probability that none of the events obtain is at least $1 - \sum p_i$. In statistics, this is known as *Bonferroni correction*.

[5]The sub-probability requirement is equivalent to the terms $\ln(1/p_i)$ being code-words lengths of a uniquely decodable code, as will be explained in Chapter 6.

The empirical Rademacher complexity of class $\mathcal{H}$ is defined as[6]

$$\hat{R}^n(\mathcal{H}) := \mathbb{E}_{\sigma^n \sim \mathrm{Uni}(\{\pm 1\}^n)} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right) \;\middle|\; x_1, \ldots, x_n \right] \quad , \tag{1.6}$$

where the expectation is taken over independent uniform $\{\pm 1\}$-valued Rademacher variables $\sigma_1, \ldots, \sigma_n$ representing randomly chosen labels, and the training instances $x_1, \ldots, x_n$ are considered fixed. The (expected) Rademacher complexity is defined as

$$R^n(\mathcal{H}) := \mathbb{E}_{x^n} \left[\hat{R}^n(\mathcal{H}) \right] \quad ,$$

where the expectation is now taken over $x_1, \ldots, x_n$. The Rademacher complexity has the following properties that make it an intuitively acceptable measure of complexity: (*i*) For a singleton class, the Rademacher complexity equals zero; (*ii*) If the class is rich enough to represent almost any configuration of the labels, the supremum in (1.6) becomes almost unity for most sequences of the Rademacher variables $\sigma_1, \ldots, \sigma_n$, and hence the Rademacher complexity of such a class is high; (*iii*) Duplicate hypotheses in the class do not affect the complexity.

Theorem 1.3 (Rademacher bound [4, 10]) *With probability at least* $1 - \delta$ *we have:*

$$\mathcal{E}(h) < \mathcal{E}^n_{\mathrm{emp}}(h_i) + 2R^n(\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{2n}} \quad \textit{for all } h \in \mathcal{H} \ .$$

It may seem problematic that the bound depends on an unknown quantity $R^n(\mathcal{H})$. However, Rademacher complexity can be approximated by the quantity inside the expectation (1.6) because this quantity is closely concentrated around its expectation (with respect to both the Rademacher variables and the training instances), see e.g. [4, Thm. 11].

If there are several hypothesis classes, the union bound can be applied in conjunction with the VC or Rademacher bounds to obtain bounds that hold for all hypotheses in all hypothesis classes at the same time. Since these bounds depend on the complexity of the hypothesis class, they are tighter for some hypotheses than for others, even though the basic bounds of Sec. 1.1 are the same for all hypotheses. Minimization of the error bound is known as the Structural Risk Minimization (SRM) principle [121], Fig. 1.2.

[6] The definition of the various Rademacher quantities varies. For instance, Bartlett and Mendelson [4] use a definition with the sum in (1.6) replaced by its *absolute value*, and multiplied by two. However, the proof of Theorem 1.3 they give does not require the absolute values. (There is a error in [4]: the last two formulas in Appendix B of the paper should be multiplied by two which removes the denominator 2 from their Theorem 5.1b.)

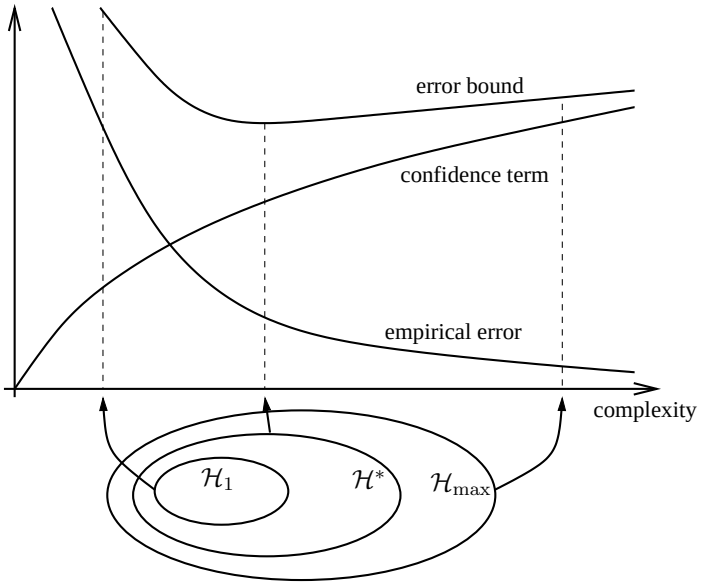

Figure 1.2: Structural Risk Minimization (SRM) principle (adapted from [121]). The error bound is a sum of the empirical error and an additional confidence term that increases with complexity of the hypothesis class $\mathcal{H}_k$. The SRM principle chooses the class $\mathcal{H}^*$ that minimizes the bound.

1.3 Discussion

Theorems 1.2 and 1.3 suggest that in order to minimize an upper bound on the generalization error, a hypothesis selection procedure should not only minimize the empirical error, but also penalize for complexity of the hypothesis. This complexity can be measured either directly in terms of the code length $\ln(1/p_i)$ for coding the hypothesis h_i, or indirectly through the complexity of the hypothesis class via $R^n(\mathcal{H})$ or related quantities. Starting from a very large set of hypotheses, for which the complexity penalty is exceedingly large, the SRM approach is to 'carve up' the hypothesis space into subsets of increasingly complexity. In the fortunate case that a relatively small subset exists that contains a hypothesis that has small empirical error, the resulting error bound is significantly tighter than would be obtained by the treating all hypotheses on an equal footing and using a single bound for the whole hypothesis space.

Chapter 2

Regression Estimation with the EM Algorithm

It is remarkable how much in statistics can be achieved by *linear* methods. Consider for instance, the problem of regression estimation. While the dependent variable y may depend on the regressor variable(s) x in a complex, non-linear way, a reasonable approximation may often be achieved by including non-linear transformations of the regressor variables in the model. Thus, for instance, the quadratic model $y = \beta_0 + \beta_1 x + \beta_2 x^2$, while non-linear in x becomes linear once the regressor x^2 is introduced. In so called *kernel methods* this idea, carried out to the extreme, yields universally flexible models which can still be computationally manageable, see [113]. In this chapter we present a method for handling partially observed data in linear regression, and its application to mobile device positioning. The work has been published in Paper 1.

Let $\mathbf{X}$ denote the *regressor* (or *design*) *matrix*:

$$\mathbf{X} := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,k} \end{pmatrix} ,$$

where the first column is often assumed to consist of all ones in order to allow constant translations like the term β_0 in the quadratic model above. Letting the column vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ (the superscript T stands for transpose) define the observed sequence of dependent variables, the linear regression model becomes

$$\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \mathbf{y} \ ,$$

where $\boldsymbol{\beta}$ is a column vector of coefficients, $\boldsymbol{\epsilon}$ is an i.i.d. sequence of error terms which are assumed Gaussian with zero mean and variance σ^2. The density of $\mathbf{y}$ is then

$$f(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2}{2\sigma^2}\right) \ , \tag{2.1}$$

where $\boldsymbol{\theta}$ denotes the pair $(\boldsymbol{\beta}, \sigma)$, and $\|\cdot\|^2$ denotes the squared Euclidean norm, i.e., the sum of squares. For fixed regressor matrix $\mathbf{X}$ and observation sequence $\mathbf{y}$, we can consider (2.1) as a function of $\boldsymbol{\theta}$. This function is called the (complete data) *likelihood function*.

The well-known least-squares method gives the maximum likelihood estimates of the parameters in closed form:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\,\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\,\mathbf{y} \ , \quad \hat{\sigma} = \sqrt{\frac{\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2}{n}}. \tag{2.2}$$

The case where some of the observations y_i are only partially observed is somewhat more complicated. In most cases, the maximum likelihood parameters do not have a closed form solutions, which calls for approximations.

2.1 Partial observability and the EM algorithm

We consider two types of partial observability. First, if the precision with which the observations are made is coarse, the observations are said to be *binned*: for each measurement we obtain only a lower bound $\underline{y}_i$ and an upper bound $\overline{y}_i$. For *truncated* (or censored) observations, we only obtain either a lower *or* an upper bound. Without loss of too much generality, we assume that the observations are labeled in such a way that the first m variables correspond to binned observations, and the $n - m$ other ones correspond to observations truncated from *above*, i.e., we have for them an upper bound $\overline{y}_i$.

Given a sequence of binned and truncated observations, the *incomplete-data likelihood*, $\mathcal{L}_{\mathrm{I}}$ (where 'I' stands for incomplete), is then defined as

$$\mathcal{L}_{\mathrm{I}}(\boldsymbol{\theta}) := \int_{Y_{\mathrm{obs}}} f(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})\, d\mathbf{y} \ , \tag{2.3}$$

where the range is defined by the observations:

$$Y_{\mathrm{obs}} := \left\{ \mathbf{y} = (y_1, \ldots, y_n) \ : \ \begin{array}{rl} \underline{y}_i \le y_i \le \overline{y}_i & \text{for } 1 \le i \le m; \\ y_i \le \overline{y}_i & \text{for } m+1 \le i \le n \end{array} \right\} \ .$$

Unfortunately, there is no analytic solution similar to (2.2) for maximization of the incomplete-data likelihood. In order to find parameter values that have as high incomplete-data likelihood as possible, it is possible to use local search heuristics like hill-climbing with (2.3) as the cost function. This tends to be inefficient unless there the cost function has certain properties, such as simple first and second derivatives, that allow the use of more sophisticated search algorithms than 'blind' search.

The expectation-maximization (EM) algorithm [23, 77] is a general heuristic for finding approximations of maximum likelihood parameters in missing-data situations. In the EM algorithm the parameters are first initialized to some values, $\boldsymbol{\theta}^{(0)}$, after which new values, $\boldsymbol{\theta}^{(1)}$, are found by maximizing the *expected* complete-data log-likelihood, the expectation being taken over $\mathbf{y} \sim f(\cdot \mid Y_{\text{obs}}, \mathbf{X}, \boldsymbol{\theta}^{(0)})$. Conditioning on Y_{obs} simply restricts the possible value to the set Y_{obs}:

$$f(\mathbf{y} \mid Y_{\text{obs}}, \mathbf{X}, \boldsymbol{\theta}^{(0)}) := \frac{f(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}^{(0)})}{\int_{Y_{\text{obs}}} f(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}^{(0)})\, d\mathbf{y}} \ .$$

The new values $\boldsymbol{\theta}^{(1)}$ are then taken as the initial point, and the process is repeated, usually until convergence. Letting $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ denote the expected log-likelihood, each iteration is then characterized by

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) := \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{y} \sim f(\cdot \mid Y_{\text{obs}}, \mathbf{X}, \boldsymbol{\theta}^{(t)})} \ln \mathcal{L}(\boldsymbol{\theta}) \ . \quad (2.4)$$

It can be shown that we have for all t the inequality

$$\mathcal{L}_{\text{I}}(\boldsymbol{\theta}^{(t)}) \leq \mathcal{L}_{\text{I}}(\boldsymbol{\theta}^{(t+1)}) \ ,$$

i.e., the likelihood never decreases during an iteration. Moreover, in typical cases, the algorithm converges to a local maximum of the likelihood function [23, 77].

It turns out that in the linear–Gaussian regression model with partially observed values, the estimators (2.2) derived for the complete-data case can still be applied, although indirectly. Namely, to obtain the estimate $\boldsymbol{\beta}^{(t+1)}$, we can simply evaluate the expectation of $\mathbf{y}$, and apply (2.2) with the expected value in place of $\mathbf{y}$. In order to obtain $\sigma^{(t+1)}$, it is also necessary to evaluate the expectation of $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(t)}\|^2$. For details, see Paper 1. In fact, this observation holds generally for all exponential family models [1], including the linear–Gaussian regression model as a special case: the maximization of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ is effectively achieved by using the same formula as in the complete-data case with the expected values of the sufficient statistics

plugged in place of their (unobserved) actual values [23]. It is important to note that this is not in general the same as 'imputation', i.e., estimating missing entries and using the estimates as they were real observations.

2.2 Signal propagation modeling and positioning

In the present work, the motivation to study linear regression with partially observable data comes from signal propagation modeling. When the signal strength on various channels is measured in a cellular telephone, the measurements are reported with finite precision. Moreover, the signal strength reading on only six to eight channels *with the strongest signal* is measured, which implies that the signal strength on the remaining channels is truncated from above. Ignoring this indirect evidence introduces selection bias in the data: for areas with low mean signal strength, only signal strength readings that are atypically high in those areas are recorded, and consequently, the mean signal strength is severely over-estimated in such areas. This phenomenon is in fact present in many observational settings where the strength of a signal is measured in a way or another.

A signal propagating freely in all directions in three dimensions attenuates in the second power of the distance, following inversely the area of a three dimensional sphere. Taking into account the path reflecting from the surface of the earth usually results in steeper attenuation due to interference, approximated in many cases by the so called *fourth-power attenuation model*, see [95]. Converting the received power p_r from units of milliwatt (mW) to units of decibel milliwatt (dBm) by

$$p_r[\text{dBm}] = 10 \times \log_{10} p_r[\text{mW}] \ ,$$

turns both the second-power and fourth-power attenuation models into the form $p_r[\text{dBm}] = \beta_0 + \beta_1 \log d$, where d is the distance from the transmitter, β_0 is a constant, and β_1 equals -20 for the second-power and -40 for the fourth-power model. In practice, the best coefficient of attenuation depends on the environment, and can be found empirically from observational data.

In Paper 1, we present a propagation model with three coefficients: a constant term, the coefficient of the log-distance term, and an additional direction-dependent factor. Including the logarithm of the distance in the model as a regressor still retains linearity of the model. Estimation of the parameters is done from partially observed data by the EM algorithm. To illustrate the method, Fig. 2.1 shows a simulation with 66 observations. In the bottom display 29 of the observations are truncated from above. By comparing the estimated signal attenuation curves in the two displays, it

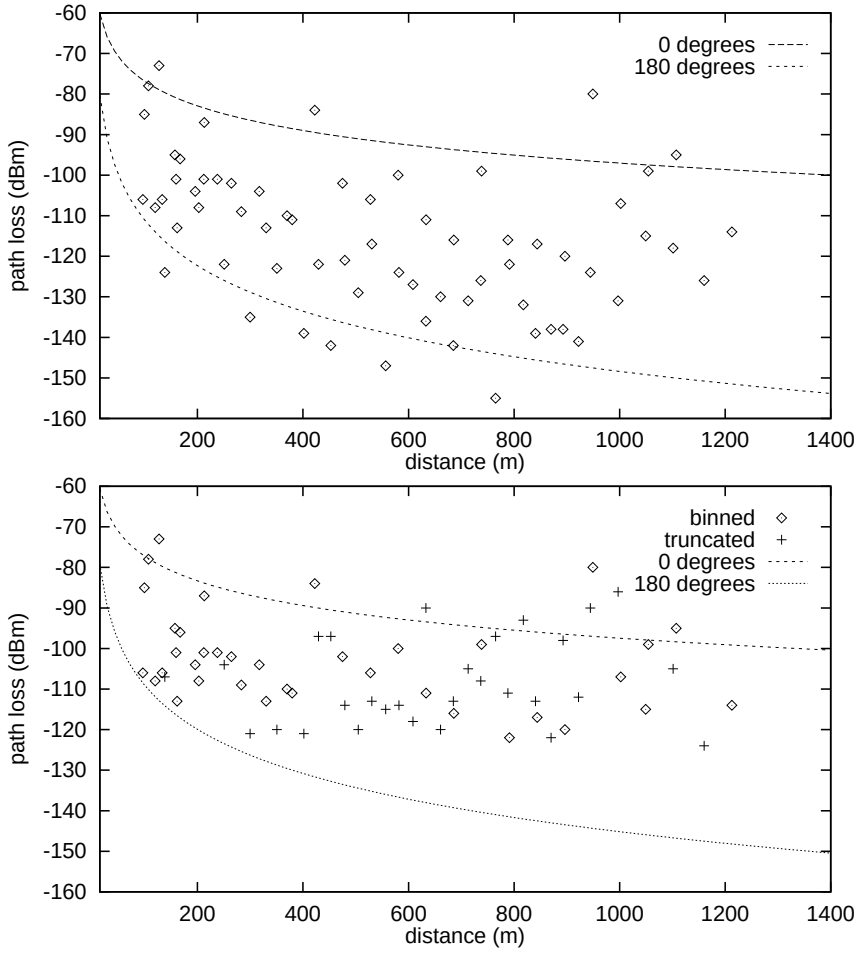

Figure 2.1: An example of signal attenuation curves estimated from fully observed (top) and partially observed (bottom) signal measurements by the EM algorithm. Diamonds ($\diamond$) indicate fully observed or binned measurements with one dBm precision (dBm = decibel milliwatt), and pluses (+) indicate upper bounds of truncated observations. The regressors in the model are the logarithm of the distance (distance on x-axis), and an additional direction-dependent factor. The two curves show the estimated mean in the direction of transmission (0°) and to the opposite direction (180°). For details, see Paper 1.

can be seen that the effect of partial observability is only marginal. Also, it can be seen from the bottom display that using only the non-truncated observations would lead to over-estimation since the measurements with weak signal tend to be truncated.

Once the parameters have been estimated, the propagation model can also be used for positioning, i.e., estimating the location of a mobile device based on received signal strength readings. The idea is to find a location in which the probability of the observed measurements is maximized, or to find the expectation of the location given the observations, see Paper 1. Figure 2.2 demonstrates the resulting errors in a simulation experiment. In the experiment the proposed method was compared to (a simplified version of) the common 'Cell-ID' method, where the location of the transmitter with the strongest received signal is used as a location estimate.

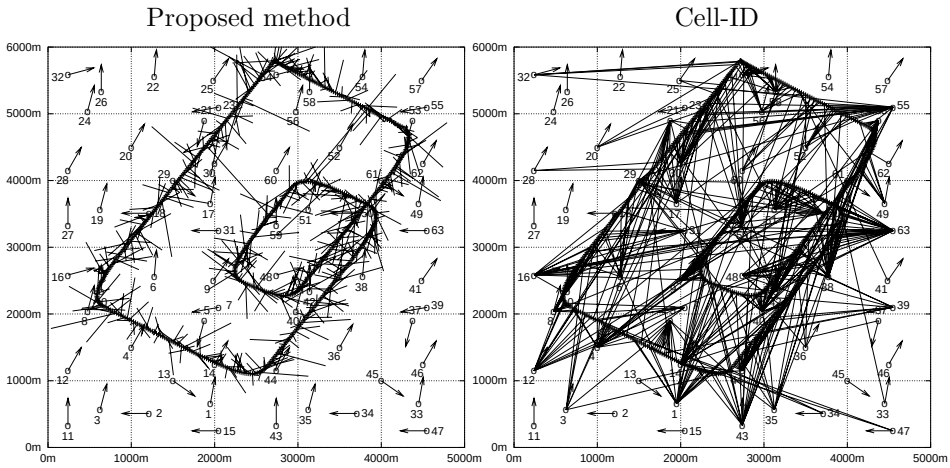

Figure 2.2: Comparison of the method proposed in Paper 1 to the Cell-ID method. A hypothetical network layout is shown in the background: a 5km × 5km area is covered by a dense network of directed transmitters, indicated by small arrows and numbers 1–64. The errors of each method are shown with lines connecting the true trajectory to the estimated location. The errors are clearly larger in the panel on the right.

Chapter 3

Generalization to Unseen Cases

> " Hence, learning is not only a question of remembering but also of *generalization to unseen cases* " [97, italics original].

One often encounters the association of the term 'generalization' to 'unseen cases' in machine learning literature. Despite the emphasis on unseen cases, such comments are invariably followed by analysis of standard generalization error. In the standard setting the test cases are i.i.d. according to the same distribution from which the training set D is sampled, which means that some of the test cases may have been already seen in the training set. In this chapter we refer to the standard generalization error as the *i.i.d. error*:

$$\mathcal{E}_{\text{iid}}(h) := \Pr[h(x) \neq y] \ .$$

If the hypothesis is chosen after seeing the training set, a more appropriate measure of generalization to unseen cases is obtained by restricting the test cases to those not already seen in the training set. This is especially true when there is little noise (stochasticity) in the outputs: then there is not much interest in the performance on the already seen instances which can simply be memorized. Restricting to the as yet unseen instances yields the *off-training-set error* [131]:

$$\mathcal{E}_{\text{ots}}(h, D) := \Pr[h(x) \neq y \mid x \notin \mathcal{X}_D] \ ,$$

where $\mathcal{X}_D \subset \mathcal{X}$ is the set of x-values occurring in the training set. If the probability of the event $x \notin \mathcal{X}_D$ is zero, the off-training-set error is undefined.

It can be argued that in many cases the instance space $\mathcal{X}$ is continuous, and that therefore, with probability one, all cases are distinct and the two error measures coincide anyway. However, it is not the continuity of the

instance space but the continuity of the distribution P that guarantees this, and as far as the distribution-free setting (see p. 4) is concerned, this cannot be taken for granted.

The off-training-set error may in some situations behave quite differently from the i.i.d. error, as demonstrated by the No Free Lunch (NFL) theorem(s) of Wolpert [131, 132, 133], see also [112, 26]. Informally stated, the NFL theorem asserts that under a uniform prior distribution on the generating distribution P, the expected off-training-set error of any learning algorithm is exactly one half. In this sense, no algorithm is better than random guessing. It is also claimed that:

1. " If we are interested in the error for [unseen cases], the NFL theorems tell us that (in the absence of prior assumptions) [empirical error] is meaningless. " [132]

2. " Unfortunately, [the tools of statistical learning theory] are ill-suited for investigating off-training-set behavior. " [133]

In Paper 2 we show that while the NFL theorem itself is mathematically valid, both of the above two claims are incorrect. This is done by presenting a method for constructing data-dependent, distribution-free off-training-set error bounds.

3.1 Missing mass

Suppose that we are modeling the distribution of words in a long sequence which is revealed sequentially from the beginning towards the end. At any time, it is possible to estimate the distribution of the words in the sequence by, for instance, the empirical distribution of words appeared so far, which maximizes the likelihood of the observed initial part of the sequence. The problem with the maximum likelihood method is that the empirical distribution assigns zero probability to all unseen words. In language modeling the remaining probability is called *missing mass*, see [76], not [61].

Definition 3.1 (sample coverage, missing mass) *Given a training set D, the* sample coverage $p(\mathcal{X}_D)$ *is the probability that a new X-value appears in D: $p(\mathcal{X}_D) := \Pr[X \in \mathcal{X}_D]$. The remaining probability, $1-p(\mathcal{X}_D)$, is called the* missing mass.

Good-Turing estimators [36], originated by Irving J. Good, and Alan Turing, are widely used in language modeling to estimate the missing mass and related quantities. It can be shown that Good-Turing estimators give

good (albeit suboptimal) estimates of the missing mass and certain other quantities in an unknown alphabet setting [91]. The known small bias of the estimators, together with bounded rates of convergence, can be used to obtain lower bounds for the missing mass, or equivalently, upper bounds on the sample coverage [75, 74].

Theorem 3.1 (Good-Turing bound [75]) *For any $0 \leq \delta \leq 1$, with probability at least $1 - \delta$:*

$$p(\mathcal{X}_D) = \mathcal{O}\left(\frac{r}{n} + \log\left(\frac{3n}{\delta}\right)\sqrt{\frac{\log(3/\delta)}{n}}\right) \ ,$$

where n is the sample size, and $0 \leq r \leq n$ is the number of instances in a random sample D with non-unique x-value[1].

The bound depends on the number of repetitions r which is a random variable determined by the sample D. In Paper 2, we state a new bound that admits the following closed-form version:

Theorem 3.2 *For any $0 \leq \delta \leq 1$, with probability at least $1 - \delta$:*

$$p(\mathcal{X}_D) = \mathcal{O}\left(\sqrt{\frac{r\log n}{n} + \frac{\log(4/\delta)}{n}}\right) \ ,$$

where n is the sample size, and r is the number of instances with non-unique x-value.

Neither of the bounds of Thms. 3.1 and 3.2 dominates the other. In order to see how they relate to each other, consider fixed δ, and increasing n. The G-T bound behaves as $\mathcal{O}(r/n + \log n/\sqrt{n})$. Our bound behaves as $\mathcal{O}(\sqrt{c + r\log n}/\sqrt{n})$, where c is a constant. We can separate three cases, depending on whether r is fixed or not:

1. For fixed $r = 0$, our bound yields $\mathcal{O}(1/\sqrt{n})$.
2. For fixed $r > 0$, our bound yields $\mathcal{O}(\sqrt{\log n/n})$.
3. For $r = \Theta(n)$, our bound becomes greater than one.

These observations hold also for the non-asymptotic version given in Paper 2. In the first two cases, our bound is asymptotically better than the G-T bound. In the third case, i.e., r growing linearly in n, our bound

[1] For instance, if the x-values in the training set are $(1, 3, 4, 1, 2)$, then $n = 5$ and $r = 2$.

becomes trivial (greater than one), but the G-T bound converges to r/n. In theory, if the data is sampled i.i.d. from some distribution P, then by the law of large numbers, with probability one, either the first or the last of the above three cases obtains. However, the asymptotic behavior does not always determine the practical utility of the bounds. In the pattern recognition context, where the sample size is modest compared to language modeling, our lower bound is more useful than the G-T bound even in cases where $r > 0$, as described in the next section.

3.2 Off-training-set error bounds

The missing mass, or the sample coverage, can be used to bound the difference between off-training-set error and i.i.d. error.

Lemma 3.1 *For all hypotheses h, and all training sets D such that $p(\mathcal{X}_D) < 1$, we have*

$$a) \quad |\mathcal{E}_{\text{ots}}(h, D) - \mathcal{E}_{\text{iid}}(h)| \leq p(\mathcal{X}_D) \ , \quad \textit{and}$$

$$b) \quad \mathcal{E}_{\text{ots}}(h, D) - \mathcal{E}_{\text{iid}}(h) \leq \frac{p(\mathcal{X}_D)}{1 - p(\mathcal{X}_D)} \mathcal{E}_{\text{iid}}(h) \ .$$

Lower bounds on the missing mass, together with Lemma 3.1a, give data-dependent bounds on the difference between the off-training-set and i.i.d. errors. For instance, Thm. 3.2 yields the following bound.

Theorem 3.3 (off-training-set error bound) *For all $0 \leq \delta \leq 1$, with probability at least $1 - \delta$, for all hypotheses h, we have*

$$|\mathcal{E}_{\text{ots}}(h, D) - \mathcal{E}_{\text{iid}}(h)| = \mathcal{O}\left(\sqrt{\frac{r \log n}{n} + \frac{\log(4/\delta)}{n}}\right) \ ,$$

where r is the number of instances in the training set D having a non-unique x-value.

The bound implies that the off-training-set error and the i.i.d. error are entangled, thus transforming all distribution-free bounds on the i.i.d. error (Hoeffding, Chernoff, etc., see Sec. 1.1) to similar bounds on the off-training-set error. Since the bound holds for all hypotheses at the same time, and does not depend on the richness of the hypothesis class in terms of, for instance, its VC dimension. Figure 3.1 illustrates the bound as the sample size grows. It can be seen that for a small number of repetitions the

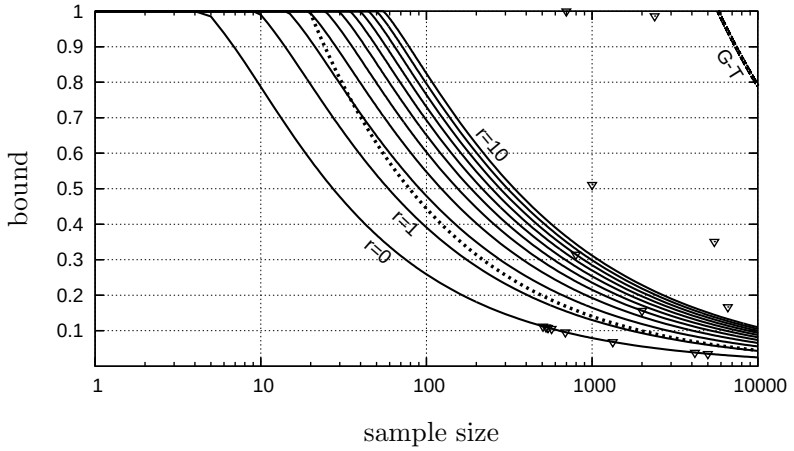

Figure 3.1: Bounds on the difference between i.i.d. and off-training-set errors. for samples with zero ($r = 0$) to ten ($r = 10$) repeated X-values on the 95 % confidence level ($\delta = 0.05$). The dotted curve is an asymptotic version for $r = 0$ given by Thm. 3.3. The curve labeled 'G-T' (for $r = 0$) is based on Good-Turing estimators (Thm. 3 in [75]). Asymptotically, it exceeds the new $r = 0$ bound by a factor $O(\log n)$. Bound for the UCI data-sets in Table 3.1 are marked with small triangles ($\bigtriangledown$). Note the log-scale for sample size.

bound is nontrivial already at moderate sample sizes. Moreover, the effect of repetitions is tolerable, and it diminishes as the number of repetitions grows. It can also be noted that the G-T bound (Thm. 3.1) is not useful for samples of size less than 10000. Table 3.1 lists values of the bound for a number of data-sets from the UCI machine learning repository [88]. In many cases the bound is about 0.10–0.20 or less.

We can now re-evaluate the two claims on p. 18. The bound we give links off-training-set error to the standard (i.i.d.) generalization error. Since it is well-known that the i.i.d. error is linked to the empirical error, this implies that empirical error is *not* meaningless to the error on unseen cases. As for second claim, the used tools are standard in statistical learning theory, what is new is their combination, which shows that these tools are *not* ill-suited for investigating off-training-set behavior.

DATA	SAMPLE SIZE	REPETITIONS	BOUND
Abalone	4177	-	0.0383
Adult	32562	25	0.0959
Annealing	798	8	0.3149
Artificial Characters	1000	34	(0.5112)
Breast Cancer (Diagnostic)	569	-	0.1057
Breast Cancer (Original)	699	236	(1.0)
Credit Approval	690	-	0.0958
Cylinder Bands	542	-	0.1084
Housing	506	-	0.1123
Internet Advertisement	2385	441	(0.9865)
Isolated Letter Speech Recogn.	1332	-	0.0685
Letter Recognition	20000	1332	(0.6503)
Multiple Features	2000	4	0.1563
Musk	6598	17	0.1671
Page Blocks	5473	80	0.3509
Water Treatment Plant	527	-	0.1099
Waveform	5000	-	0.0350

Table 3.1: Bounds on the difference between the i.i.d. error and the off-training-set error on confidence level 95% ($\delta = 0.05$). A dash (-) indicates no repetitions. Bounds greater than 0.5 are in parentheses.

Part II

The Bayesian Approach

Chapter 4

Preliminaries

The statistical learning framework of Chapter 1 is formalized in terms of classical frequentist statistics, such as fixed but unknown parameters and their estimators. The Bayesian approach to data analysis builds upon the Bayesian paradigm with its own concepts that differ, in some aspects utterly, from the frequentist ones. The central idea in Bayesianism is to use a subjective joint probability distribution to represent uncertainty in all unknown quantities, see e.g. [111, 7]. Since uncertainty is a property related to knowledge, and knowledge is always *someone's* knowledge about something, Bayesian probabilities are often subjective, although in some situations there are rather strict restrictions on what can be called *rational* beliefs. For instance, probabilities that arise in sampling scenarios, e.g., randomized experiments, are often the same independently of which interpretation, the subjectivistic or the frequentist, is assumed.

In Bayesian theory, there is no distinction between parameters and random variables, like there is in frequentist theory. Hence, in addition to sampling-related probabilities, Bayesians assign probabilities to many events that are not considered random in the frequentist framework. To emphasize this different categorization — random–fixed vs. random–known — the term 'random quantity' is sometimes used in the Bayesian context, covering both unknown random variables and quantities that a frequentist statistician would call fixed but unknown parameters. Once information is obtained that is relevant to any of such random quantities, their distribution is conditioned on this information. All inference tasks use the basic operations of probability calculus.

The Bayesian worldview, in comparison to the frequentist one, is arguably closer to our everyday conception of probability, confidence and related notions. For instance, the interpretation of frequentist confidence intervals is that prior to observing the data, the probability that the in-

terval will include the true value is high. Nothing can be said about the validity of the bound *conditional* on the observed data since all randomness is in the data. The problem is that the probability that any *given* interval contains the true value is necessarily either zero or one, but we do not know which. The interpretation of Bayesian confidence intervals (or rather, to follow the terminology, high probability intervals) is very natural: given the observed data, the true value of the estimated quantity is with high probability within the obtained range. More generally, frequentist methods deal with 'initial precision', whereas the Bayesian framework is focused on 'final precision' [6].

4.1 Bayesian inference

Although, in principle, everything in Bayesian inference is standard probability calculus, it is worthwhile to make some more specific remarks concerning the concepts and techniques that are often encountered in practice. A more detailed exposition is given in, for instance, [7].

In Bayesian statistical inference, the probability distribution over observables is often constructed in two parts. First, a parametric model is assumed that gives a distribution over the observables conditional to one or more parameters. The unknown parameters are modeled by a prior distribution. The joint distribution of a sequence of observable variables, $x^n = (x_1, \ldots, x_n)$, and the parameters, θ, then factorizes as

$$p(x^n, \theta) = p(x^n \mid \theta)\, p(\theta) \ . \tag{4.1}$$

If the components of x^n are i.i.d. given θ, then for all $x^n \in \mathcal{X}^n$ we have

$$p(x^n \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta) \ .$$

The distribution of the observables is obtained from the joint distribution of x^n and θ by *marginalization*:

$$p(x^n) = \int_{\Theta} p(x^n \mid \theta)\, p(\theta)\, d\theta \ , \tag{4.2}$$

where Θ is the parameter domain. Integrals of this form are often called *Bayes mixtures*.

From a purely subjectivistic Bayesian perspective, all uncertainty is epistemic, due to our ignorance, and does not exist in any objective sense (for an extreme view, see [53]). In this light, the status of parameters,

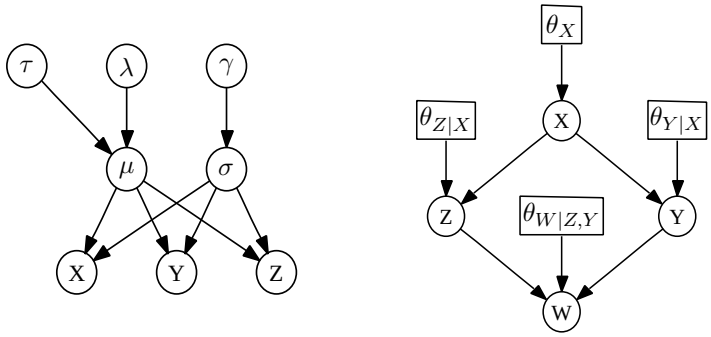

Figure 4.1: Two examples of graphical models. The graph on the left is a typical hierarchical (or multilevel) model with hyper-priors for parameters μ and σ. The graph on the right is a Bayesian network; the parameters of the conditional probability tables for each node, $\theta_X, \theta_{Z|X}$, etc. are often omitted from the figure.

and terms like $p(\theta)$, is problematic. However, factorizations like (4.2) can also be obtained without explicit reference to parameters from the distribution of observables via the weaker assumption of *exchangeability* of the observables [30, 7].

> **Graphical Models:** The prior–likelihood model is sometimes hierarchical. For instance, the parameter prior may be expressed as a mixture of the form (4.2), with θ in place of x, and a *hyper-parameter* α in place of θ. The term $p(\alpha)$ is then the hyper-prior that may be defined as a function of hyper-hyper-parameters, etc. Complex hierarchical models are conveniently expressed in terms of *graphical models*, as in Fig. 4.1. In principle, there is nothing Bayesian about graphical models, and many graphical models are used in non-Bayesian ways; for instance, Kalman filters, Markov random fields, and hidden Markov models can all be viewed as graphical models. However, the interpretation of especially hierarchical models is much more straightforward in the Bayesian context.

The conditional distribution of the parameters given data D is obtained by *conditionalization* via Bayes's rule:

$$p(\theta \mid D) = \frac{p(D, \theta)}{p(D)} = \frac{p(D \mid \theta)\, p(\theta)}{p(D)} \quad ,$$

which is sometimes expressed using the proportionality symbol '$\propto$' as

$$\begin{array}{ccccc} p(\theta \mid D) & \propto & p(D \mid \theta) & \times & p(\theta) \\ \textit{posterior} & \propto & \textit{likelihood} & \times & \textit{prior}. \end{array}$$

If a family of models, $\mathcal{M} = \{M_1, M_2, \ldots\}$, is contemplated, the meaning of the parameters may depend on the model, and we write $\theta_k \in \Theta_k$ for the parameters of model $M_k \in \mathcal{M}$. In situations where the task is to select one of the models, a natural model selection criterion is to maximize the posterior probability of the model given data D:

$$\begin{aligned} p(M_k \mid D, \mathcal{M}) &\propto p(D \mid M_k)\, p(M_k \mid \mathcal{M}) \\ &= \left(\int_{\Theta_k} p(D \mid \theta_k, M_k)\, p(\theta_k \mid M_k)\, d\theta_k \right) p(M_k \mid \mathcal{M}) \ , \end{aligned} \tag{4.3}$$

where $p(M_k \mid \mathcal{M})$ is a model prior. The important term $p(D \mid M_k)$ is called *marginal likelihood* (or *evidence*).

The *predictive distribution* of x given D under model M_k is given by

$$p(x \mid D, M_k) = \int_{\Theta_k} p(x \mid \theta_k, D, M_k)\, p(\theta_k \mid D, M_k)\, d\theta_k \ .$$

If x and D are independent given θ_k, then $p(x \mid \theta_k, D, M_k) = p(x \mid \theta_k, M_k)$, and the predictive distribution becomes

$$p(x \mid D, M_k) = \int_{\Theta_k} p(x \mid \theta_k, M_k)\, p(\theta_k \mid D, M_k)\, d\theta \ , \tag{4.4}$$

where the data appears only in the posterior distribution of θ_k. This gives the predictive distribution (4.4) as a mixture of the form (4.2).

Computational resources allowing, it is generally better to 'marginalize out' both the parameters *and the models*. This gives a predictive distribution conditioned on D:

$$p(x \mid D, \mathcal{M}) = \sum_{M_k \in \mathcal{M}} p(x \mid D, M_k)\, p(M_k \mid D, \mathcal{M}) \ , \tag{4.5}$$

where $\mathcal{M}$ is the considered family of models. The sense in which model averaging is better than model selection is discussed in [51]: the central point is that predictions based on a single model tend to be over-confident due to ignorance of model uncertainty.

4.2 Bayesian Occam's razor

Figure 4.2 illustrates an Occam's razor effect implicit in the marginal likelihood term [117, 68]. For a complex model, there are parameter configurations yielding high conditional probability $p(D \mid \theta_k, M_k)$ for almost all

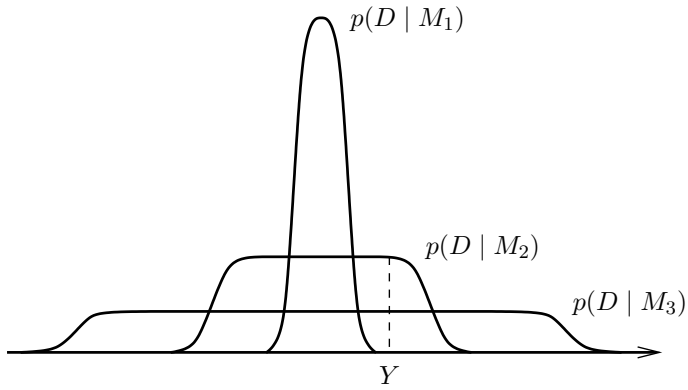

Figure 4.2: Bayesian Occam's razor (adapted from [68, 96]). The marginal likelihood of three models, M_1, M_2, M_3, plotted for all possible data-sets ordered along a one-dimensional representation. A simple model, M_1, gives high probability to only few data-sets. A complex model, M_3, covers almost all data, but due to normalization, gives relatively small probability to each data-set. For data Y, model M_2 is "just right".

data-sets. However, the requirement that the total mass over all D is always equal to one, still implies that the marginal likelihood $p(D \mid M_k)$ has to be low on average, which pushes the curve down for all data-sets. In comparison, a simpler model, e.g., a linear model instead of a non-linear one, may give very high marginal likelihood to only some very special data-sets; for instance, linear models fit data-sets with roughly 'linear' structure.

The above can be made formal by considering the asymptotics of the marginal likelihood. For parametric models, there are various different approximations, differing in terms of the regularity conditions they impose on the models and/or the generating distribution, see e.g. [115, 18, 90, 63]. A typical result is the following.

Theorem 4.1 (Evidence approximation) *Under regularity conditions, the logarithm of the marginal likelihood under a k-parameter model M_k is approximated by*

$$\begin{aligned} \ln p(D \mid M_k) = \ln p(D \mid \hat{\theta}_k(D), M_k) - \frac{k}{2} \ln \frac{n}{2\pi} \\ + \ln p(\hat{\theta}_k(D) \mid M_k) - \frac{1}{2} \ln \det I(\hat{\theta}_k(D)) + o(1) \ , \end{aligned} \tag{4.6}$$

where $\hat{\theta}_k(D)$ denotes the maximum likelihood parameters for data D, and $I(\cdot)$ is the Fisher information matrix, and the remainder term $o(1)$ goes to zero as $n \to \infty$.

The regularity conditions are usually related to smoothness of the likelihood and the prior. For details, see the aforementioned papers. Retaining only the (asymptotically) most significant terms in (4.6) gives the well-known BIC model selection criterion [115]:

$$\mathrm{BIC}(D,k) := \ln p(D \mid \hat{\theta}_k(D), M_k) - \frac{k}{2} \ln n \ , \tag{4.7}$$

which is sometimes expressed in a form where the terms are multiplied by two and negated, i.e., $-2 \ln p(D \mid \hat{\theta}_k(D), M_k) + k \ln n$. In nested model families, the first term of Eq. (4.7) grows and the second term becomes smaller (more negative) as k is increased, which demonstrates the Occam's razor effect in an asymptotic manner. For non-asymptotic experimental results, see e.g. [96, 85]. It should be noted that for practical purposes, the accuracy of the BIC approximation is very rough, and more accurate analytic approximations are to be preferred. The use of stochastic approximations, such as Monte Carlo sampling [12, 16, 35], has also become very popular in model selection as well as other tasks.

4.3 Principle of maximum expected utility

To convert beliefs and utilities into decisions, an inference mechanism needs to be complemented with a decision principle. In the Bayesian context, the natural principle is that of maximum expected utility. Utility can be equated with negative loss, so the principle could be phrased, using terminology of the previous section, the principle of minimum expected loss (or minimum risk). The essential difference between expected loss of a hypothesis as defined in (1.1) and the expected loss in the Bayesian sense is that in the latter, the expectation is taken under the predictive distribution of the random inputs, conditioned on the training set D:

$$\mathcal{E}(h \mid D) := \mathbb{E}_{(x,y)} \left[\ell(y, h(x)) \mid D \right] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(x)) \, p(x, y \mid D) \, dx \, dy \ .$$

Using this notation, the frequentist setting would be obtained by replacing the observed training set D by the unknown generating distribution P.

The hypothesis (or more generally, decision) that minimizes the expected risk

$$h_{\mathrm{Bayes}}(D) := \operatorname*{arg\,min}_{h \in \mathcal{H}} \mathcal{E}(h \mid D) \tag{4.8}$$

is called the *Bayes optimal solution* or the *Bayes act*. The Bayes optimal solution to the three problems in Sec. 1, p. 4, is immediate since the predictive distribution $p(x, y \mid D)$ is known, inasmuch as subjective probabilities

can be said to be 'known'. The solutions stated in connection to the said problems are simply applied with the predictive distribution in place of the generating distribution P: in classification, the best label is given by $\arg\max_y p(y \mid x, D)$, etc. The Bayesian approach to machine learning and related areas is then largely the task of constructing new likelihood–prior models and computational methods for minimizing expected loss under them. Popular examples include the naïve Bayes classifier [71, 81, 25] and other Bayesian network classifiers [31].

4.4 Discussion

There are several different justifications for Bayesianism. First, one can derive the rules of probability calculus from a set of axioms on handling degrees of belief that can be claimed to appeal to common sense [94, 20, 45, 53]. All rational inference in the sense of conforming to such axioms can be shown to be equivalent to Bayesian inference using some prior distribution. On the other hand, rational *behavior* can be characterized by a set of axioms that can be argued to be compelling [30, 111]. In particular, any set of beliefs incoherent with probability theory can be used to construct a combination of bets that yields loss for all possible outcomes, a so called Dutch book [94, 30] (see also [52, 120]).

One of the most criticized issues in the Bayesian approach is related to priors. Even if there is a consensus on the use of probability calculus to update beliefs, wildly different conclusions can be arrived at from different states of prior beliefs. While such differences tend to diminish with increasing amount of observed data, they are a problem in real situations where the amount of data is always finite. Further, it is only true that posterior beliefs eventually coincide if everyone uses the same set of models and all prior distributions are mutually continuous, i.e., assign non-zero probabilities to the same subsets of the parameter space ('Cromwell's rule', see [67]; these conditions are very similar to those guaranteeing consistency [8]). As an interesting sidenote, a Bayesian will always be sure that her own predictions are 'well-calibrated', i.e., that empirical frequencies eventually converge to predicted probabilities, no matter how poorly they may have performed so far [22].

It is actually somewhat misleading to speak of the aforementioned criticism as the 'problem of priors', as it were, since what is meant is often at least as much a 'problem of models': if a different set of models is assumed, differences in beliefs never vanish even with the amount of data going to infinity. Hence, compared to the choice of priors, much stronger subjec-

tivity is exercised in the choice of models. However, this point tends to be forgotten in arguments against the Bayesian approach since it concerns just as much any approach, including the frequentist one.

Chapter 5

Discriminative Bayesian Network Classifiers

The situation in which data is generated by a model outside the set of models under consideration — or, in more subjectivistic terms, behave as if they were generated so — is called *misspecification*. While a subjectivistic Bayesian is sure that this is never the case [22], more pragmatic considerations suggest that it is useful to be prepared for the worst. This is called the $\mathcal{M}$-open view [7]. For instance, discriminative (or supervised) learning that targets directly the prediction task at hand gives sometimes significantly better results than standard 'generative' (or unsupervised) methods [110, 89, 54]. In this chapter we discuss discriminative learning of Bayesian network classifiers, see e.g. [38, 59, 39]. This work has been published in Paper 3.

5.1 Prediction under misspecification

Decisions following the principle of maximum expected utility (Sec. 4.3), are by definition optimal in the expected sense under the assumed model. It is also important to consider how robust this approach is with respect to misspecification: what happens when data is sampled from one distribution and the decisions are derived using another distribution. While such a sampling-oriented setting is inherently non-Bayesian, we can alternatively think of the 'true' distribution as someone else's subjective distribution under which our decisions are evaluated. Such considerations are relevant to group decision making (see e.g. [8] and [7, Ch. 6]).

The situation is strongly affected by whether the generating distribution is inside or outside the assumed model. In the case where the generating

distribution is inside the model, Bayesian methods are consistent under rather weak regularity conditions (for instance, smoothness of the model and prior, see e.g. [8, 114, 18]), and will eventually yield optimal predictions that match the generating distribution. Consequently, the derived decisions also converge to the optimal ones. In contrast, when the generating distribution is outside the model, i.e., in the case of misspecification, the posterior distribution cannot be consistent in general, and the quality of the decisions is not guaranteed. In this case, it is interesting to compare the performance of predictors to the best predictor achievable with the given model, i.e., the best predictor among the set of Bayes acts derived from the distributions in the model.

5.2 Bayesian network classifiers

Bayesian networks are probabilistic graphical models (see [92, 65]) that are composed of two parts: a directed acyclic graph (DAG) that determines the independence–dependence relations among the relevant variables, and a set of associated conditional probability distributions. The distributions are usually defined through a family of parametric models and a set of parameters. In applications, the primary interest is sometimes in discovering the independence–dependence relations, i.e., the DAG, and quantitative probability assessments are of secondary interest; consider, for instance, discovery of gene regulatory networks [32]. Here we consider the task of prediction, focusing primarily on parameter learning.

A Bayesian network defines a joint probability distribution over a set of domain variables, $x_0, \ldots, x_k$, by a DAG, $\mathcal{B}$, and a set of local probability distributions as follows:

$$p(x_0, \ldots, x_k \mid \theta^{\mathcal{B}}, \mathcal{B}) = \prod_{i=0}^{k} p(x_i \mid \mathrm{pa}_i^{\mathcal{B}}, \theta^{\mathcal{B}}, \mathcal{B}) \ , \tag{5.1}$$

where $pa_i^{\mathcal{B}}$ denotes the set of immediate predecessors (parents) of variable x_i in the graph $\mathcal{B}$, and $\theta^{\mathcal{B}}$ denotes the parameters defining the conditional distributions. For simplicity, we assume that all variables are discrete, and that the conditional distributions are multinomial. In this case the model can be parameterized by parameters of the form $\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}}$ for all $x_i \in \mathcal{X}_i$ and $\mathrm{pa}_i^{\mathcal{B}} \in \mathcal{X}_{p_1(i)} \times \ldots \times \mathcal{X}_{p_{m(i)}(i)}$, where $p_1(i), \ldots, p_{m(i)}(i)$ are the parents of x_i, by setting

$$p(x_i \mid \mathrm{pa}_i^{\mathcal{B}}, \theta^{\mathcal{B}}, \mathcal{B}) := \theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}} \ . \tag{5.2}$$

To extend the model to an i.i.d. training set D, let $x_{i,j}$ denote the jth realization of x_i in D. Under the i.i.d. multinomial model parameterized as (5.2), we have

$$p(D \mid \theta^{\mathcal{B}}, \mathcal{B}) = \prod_{j=1}^{n} p(x_{0,j}, \ldots x_{k,j} \mid \theta^{\mathcal{B}}, \mathcal{B}) = \prod_{j=1}^{n} \prod_{i=0}^{k} \theta_{x_{i,j}|\mathrm{pa}_{i,j}^{\mathcal{B}}} \ .$$

If the parameters are assumed independent of each other, with a Dirichlet prior, the posterior distribution of the parameters has an especially convenient form. Namely, the Dirichlet distribution is a so called *conjugate family* for the multinomial model [11, 47], which means that the posterior is also Dirichlet. In particular, the posterior mean of each parameter $\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}}$ is given by

$$\mathbb{E}_{\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}}} \left[\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}} \mid D \right] = \frac{\alpha_{x_i|\mathrm{pa}_i^{\mathcal{B}}} + n_{[x_i, \mathrm{pa}_i^{\mathcal{B}}]}}{\alpha_{\mathrm{pa}_i^{\mathcal{B}}} + n_{[\mathrm{pa}_i^{\mathcal{B}}]}} \ , \tag{5.3}$$

where $n_{[\cdot]}$ denotes the number of vectors in D that match the argument, $\alpha_{x_i|\mathrm{pa}_i^{\mathcal{B}}}$ are (hyper-)parameters of the Dirichlet prior, and

$$\alpha_{\mathrm{pa}_i^{\mathcal{B}}} := \sum_{x_i \in \mathcal{X}_i} \alpha_{x_i|\mathrm{pa}_i^{\mathcal{B}}} \ .$$

In the standard multinomial parameterization (5.2), the posterior mean equals the predictive probability of a single variable given its parents and the training data [47]:

$$p(x_i \mid \mathrm{pa}_i^{\mathcal{B}}, D, \mathcal{B}) = \mathbb{E}_{\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}}} \left[\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}} \mid D \right] \ , \tag{5.4}$$

and hence, by parameter independence and equations (5.1) and (5.4), the joint predictive distribution becomes a product of terms of the form (5.3):

$$p(x_0, \ldots, x_k \mid D, \mathcal{B}) = \prod_{i=0}^{k} p(x_i \mid \mathrm{pa}_i^{\mathcal{B}}, D, \mathcal{B}) = \prod_{i=0}^{k} \frac{\alpha_{x_i|\mathrm{pa}_i^{\mathcal{B}}} + n_{[x_i, \mathrm{pa}_i^{\mathcal{B}}]}}{\alpha_{\mathrm{pa}_i^{\mathcal{B}}} + n_{[\mathrm{pa}_i^{\mathcal{B}}]}} \ . \tag{5.5}$$

Given a Bayesian network, the corresponding *Bayesian network classifier* [31] is obtained by letting one of the domain variables be a target variable, assumed here without loss of generality to be x_0. The remaining variables, $x_1, \ldots, x_k$, are called predictor variables. Given a Bayesian network and a training set D, the predictive distribution of the target variable given the predictor variables becomes:

$$p(x_0 \mid x_1, \ldots, x_k, D, \mathcal{B}) = \frac{p(x_0, \ldots, x_k \mid D, \mathcal{B})}{\sum_{x_0' \in \mathcal{X}_0} p(x_0', x_1, \ldots, x_k \mid D, \mathcal{B})} \ . \tag{5.6}$$

For the multinomial–Dirichlet model, this can be evaluated in closed form using Eq. (5.5).

5.3 Large-sample asymptotics

From an asymptotic point of view, it is easy to see from (5.3) that the posterior means, and hence also the predictive probabilities, tend towards the empirical frequencies

$$p(x_i \mid \mathrm{pa}_i^{\mathcal{B}}, D, \mathcal{B}) \underset{n\to\infty}{\longrightarrow} \frac{n_{[x_i,\mathrm{pa}_i^{\mathcal{B}}]}}{n_{[\mathrm{pa}_i^{\mathcal{B}}]}} \ ,$$

assuming that the counts grow with the sample size. The empirical frequency is in fact the maximum likelihood estimate of the parameter $\theta_{x_i|\mathrm{pa}_i^{\mathcal{B}}}$. Consequently, the joint predictive distribution converges for all $x_0 \in \mathcal{X}_0, \ldots,$ $x_k \in \mathcal{X}_k$ to the distribution defined by the maximum likelihood parameters (see [27]):

$$p(x_0, \ldots, x_k \mid D, \mathcal{B}) \underset{n\to\infty}{\longrightarrow} p(x_0, \ldots, x_k \mid \hat{\theta}^{\mathcal{B}}(D), \mathcal{B}) \ , \tag{5.7}$$

where the maximum likelihood estimator is defined as usually:

$$\hat{\theta}^{\mathcal{B}}(D) := \arg\max_{\theta^{\mathcal{B}} \in \Theta^{\mathcal{B}}} p(D \mid \theta^{\mathcal{B}}, \mathcal{B}) \ . \tag{5.8}$$

The maximum likelihood estimate may not be unique but the result holds for all choices in the ambiguous cases. Note, however, that the convergence to empirical frequencies does not hold in general for the joint predictive distribution

$$p(x_0, \ldots, x_k \mid D, \mathcal{B}) \underset{n\to\infty}{\not\longrightarrow} \frac{n_{[x_0,\ldots,x_k]}}{n} \ ,$$

unless the data behaves as if the model were 'correct' (not misspecified). This is trivially achieved for the fully connected DAG, i.e., when all nodes are directly connected to each other. In contrast, for the empty DAG with no edges, convergence is guaranteed only in terms of the marginal distributions $p(x_i \mid D, \mathcal{B})$ for $0 \leq i \leq k$.

For Bayesian network classifiers, under mild regularity conditions (necessary to guarantee that the denominator in (5.6) grows; see Paper 3), the conditional predictive distribution is well approximated by plugging the right-hand side of (5.7) into both the numerator and denominator of (5.6). We call the resulting predictor the *ML-plug-in predictor*. Hence, the asymptotic behavior of the Bayesian predictive distribution follows that of the ML-plug-in predictor. Moreover, it is straightforward to modify the ML-plug-in predictor for supervised learning tasks.

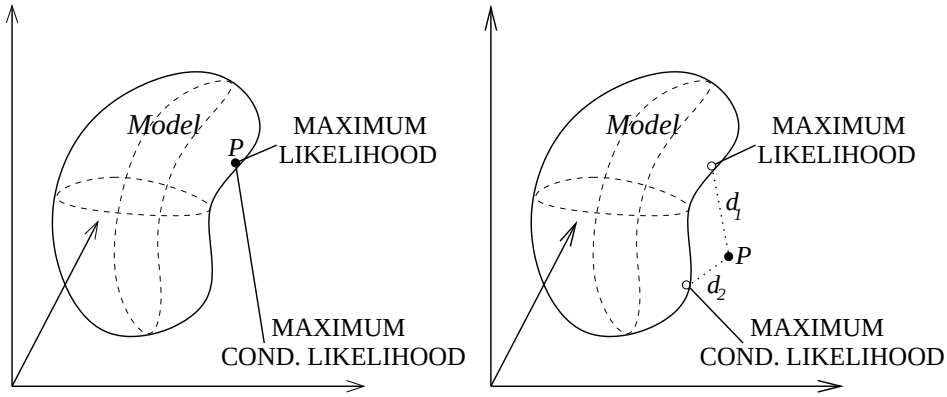

Figure 5.1: Two scenarios. *Left:* The generating distribution P is inside the assumed model. Under regularity conditions, both the ML-plug-in and MCL-plug-in predictors are consistent. *Right:* The generating distribution P is outside the model. Asymptotically, the excess risk d_1 of the conditional ML-plug-in predictions is larger than the excess risk d_2 of the MCL-plug-in predictions, measured in terms of the expected conditional log-loss (conditional KL-divergence).

5.4 Discriminative parameter learning

In the conditional density estimation task, a natural alternative to maximum likelihood estimation is to find the parameters maximizing the *conditional likelihood* (compare to (5.8)):

$$\hat{\theta}^{\mathcal{B}}_{\mathrm{cond}}(D) := \arg\max_{\theta^{\mathcal{B}} \in \Theta^{\mathcal{B}}} p(D_0 \mid D_1, \ldots, D_k, \theta^{\mathcal{B}}, \mathcal{B}) \ , \tag{5.9}$$

where for each $i \in \{0, \ldots, k\}$, $D_i = (x_{i,1}, \ldots, x_{i,n})$ denotes the sequence of the n realizations of x_i in the training set. The corresponding plug-in predictor is called the *MCL-plug-in predictor*. For conditional log-loss, the latter converges under regularity conditions to the best predictor achievable with the model, but the same cannot be said about the ML-plug-in predictor (for details, see Prop. 1 and Example 4 in Paper 3). Figure 5.1 illustrates the asymptotic behavior of the ML-plug-in and MCL-plug-in predictors in two situations, in which data is generated by a distribution P inside and outside a parameterized model, respectively.

> **Discriminative Learning or Discriminative Models?** It has recently been suggested that the term 'discriminative learning' should be abandoned, and that one should rather speak of 'discriminative models' [80, 64] (see also [44]). The idea is to consider the following double-parameterization: one set of parameters, θ, defines the conditional distribution of the target

variable(s) given the predictor variables, and another set of parameters, θ', defines the marginal distribution of the predictor variables. The standard generative model is recovered by enforcing $\theta = \theta'$. Letting the two parameter sets be independent, and maximizing the joint likelihood with respect to both θ and θ', gives the same estimate of θ, and hence the same conditional predictions, as *conditional* likelihood maximization. Note that even though generative learning is often thought to be computationally easier than discriminative learning, the transformation of a model to a discriminative version makes the two types of learning equally hard. Hence the value of the idea is not pragmatic, but foundational: it gives a principled justification to conditional likelihood maximization. It is unclear whether the idea can be extended to other optimization criteria (loss functions), such as the 0/1 loss.

Unfortunately, in contrast to the Bayesian predictive distribution and the ML-plug-in predictor, no closed form solution is available for the MCL-plug-in predictor, see [31]. Nevertheless, it has been suggested that local search heuristics, such as gradient descent, can be used to find a *local* maximum of the conditional likelihood [48].

The main theoretical contribution of Paper 3 is to show that under a simple condition on the DAG structure of a Bayesian network classifier, the conditional likelihood is a unimodal function of the parameters. In addition, in a suitable re-parameterization, the likelihood surface is in fact log-concave, and the parameter space is convex. This implies, among other things, that any local optimum is in fact necessarily *global*, and that the effective search methods developed for convex optimization and logistic regression [79] can be applied.

Definition 5.1 (Moral node) *A node in a DAG is said to be* moral *if all its parents are connected by an edge.*

Theorem 5.1 *If the DAG structure is such that after fully connecting all parents of the target variable with each other, all children of the target variable are moral, then there is a parameterization in which the conditional likelihood is a log-concave function of the parameters, and the parameter space is convex.*

Figure 5.2 shows four examples of DAGs, two of which satisfy the condition of Theorem 5.1, and two of which do not. Further positive examples include the naïve Bayes, and tree-augmented naïve Bayes [31] (for instance, the third graph in Fig. 5.2) models. Any Bayesian network can be made to satisfy the condition by adding edges, which of course increases model

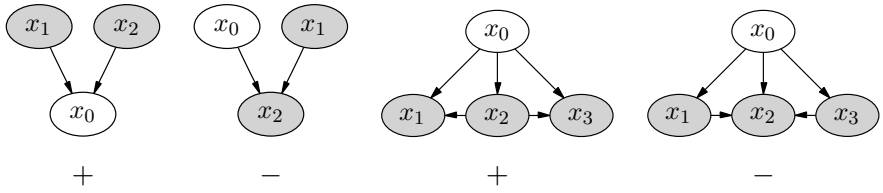

Figure 5.2: Four examples of DAGs. The target variable is x_0. The first and the third DAG satisfy the condition of Theorem 5.1 (indicated by the plus (+) sign); the second and the fourth do not.

complexity. In Paper 3 we give an explicit example (a data set) in which the second graph of Figure 5.2 induces local optima in the conditional likelihood surface. This shows that the condition in Theorem 5.1 is not superfluous. It is currently unknown if the given condition is also *necessary* for unimodality (for all data sets).

Part III

Minimum Description Length Principle

Chapter 6

Preliminaries

The Minimum Description Length (MDL) principle [98, 101] is a relatively recent framework, compared to the Bayesian and frequentist approaches. It arose from obvious difficulties of the frequentist framework to deal with the problem of over-fitting in model selection, and on the other hand, from Kolmogorov's theory of algorithmic complexity [57] and the related theory of universal prediction of Solomonoff [118]. The idea of MDL was also inspired by the earlier Minimum Message Length (MML) principle [128]. While the two principles, MDL and MML, are superficially similar, their development has been largely independent and they differ in many foundational and practical issues [5]. Most notably, (*i*) unlike MDL, MML is a Bayesian method, and (*ii*) MML selects a single hypothesis while MDL (typically) selects a model class. For comprehensive reviews on MDL, see [3, 46, 42, 43].

The three central concepts in the theory of MDL are *complexity, information*, and *noise*. Roughly, their relationship is that the total complexity in an object is the sum of the information and the noise in it. The objective of MDL is then to extract the information from a given set of data. The MDL principle itself calls for the model that minimizes the total description length:

$$\widehat{M}_{\mathrm{MDL}} := \arg\min_{M_k \in \mathcal{M}} \mathcal{L}(M_k) + \mathcal{L}(D \; ; \; M_k) \;\; , \tag{6.1}$$

where $\mathcal{L}(M_k)$ and $\mathcal{L}(D \; ; \; M_k)$ denote description (or *code*-) length of the model, and the data given the model, respectively. In many (but *not* all) practical situations the first term is ignorable in comparison to latter one, and can be omitted.

> **Code-lengths and Probabilities:** There is an important relationship between code-lengths and probabilities, implied by the Kraft-McMillan inequality [60, 78, 19]. A sequence of integers $l_1, l_2, \ldots$ can represent the

code-words lengths of a uniquely decodable code (in bits) if and only if it satisfies the inequality

$$\sum_{i=1}^{\infty} 2^{-l_i} \leq 1 \ . \tag{6.2}$$

This allows the unification of codes and (sub-)probability distributions with probabilities given by $p_i = 2^{-l_i}$. The restriction that l_i are integers is of no practical importance since strings of symbols can be encoded in a block-wise fashion so that the rounding up to the nearest integer needs to be done only at the end of each block. An analogous result can be obtained for continuous distributions by discretization. We call codes with non-integer code-word lengths satisfying (6.2) *ideal codes*. It is often convenient to express code-lengths in units of *nats* instead of bits, corresponding to the use of natural logarithm, in which case we have $p_i = \exp(-l_i)$, or equivalently, $l_i = \ln(1/p_i)$.

Using the correspondence between ideal code-lengths and probabilities, the MDL criterion (6.1) can be written as

$$\widehat{M}_{\mathrm{MDL}} := \arg\max_{M_k \in \mathcal{M}} p(M_k) \times p(D \ ; \ M_k) \ ,$$

where $p(M_k)$ and $p(D \ ; \ M_k)$ are probabilities corresponding to the code-lengths $\mathcal{L}(M_k)$ and $\mathcal{L}(D \ ; \ M_k)$, respectively. This seems to suggest that MDL and Bayesian model selection by maximization of posterior probability (4.3) are equivalent. However, the interpretation of the term $p(D \ ; \ M_k)$ is different from that of marginal likelihood in the Bayesian framework, as emphasized by the different notation (';' vs. '|'). This is not a mere terminological distinction but actually leads to practical differences in many cases, often related to the different optimality criteria (expected loss vs. worst-case relative loss) and the choice of priors in the Bayesian model.

6.1 'Ideal' vs. practical MDL

In the context of Kolmogorov complexity, the idea of decomposing the total complexity into information and noise is encapsulated by the so called Kolmogorov minimal sufficient statistic and the related 'ideal MDL' principle. In order to discuss these, we introduce some definitions related to Kolmogorov complexity; for more material, see [135, 19, 66].

A *prefix-free Turing machine* is a Turing machine whose halting programs form a prefix-free set. The *prefix-free Kolmogorov complexity* $K_U(x)$ of (a description of) an object x is defined as the length (in bits) of the

shortest program that produces x when run on a universal prefix-free Turing machine U. The definition of $K(x)$ depends on the specific Turing machine U. However, as the complexity of x increases, this dependency becomes asymptotically negligible since for any two universal machines, U and V, we have $|K_U(x) - K_V(x)| \leq c_{U,V}$ for all x, where $c_{U,V}$ is a fixed (but usually unknown) constant. The prefix-free property implies, via the Kraft-McMillan inequality (6.2), that the sum of terms $2^{-K_U(x)}$ is at most one, and hence, that there is an associated universal (sub-)probability distribution, defined by $P_U(x) := 2^{-K_U(x)}$. The dependency on the universal machine U is usually omitted from the notation, and we write $K(x) = K_U(x)$.

The standard definition of the Kolmogorov minimal sufficient statistic, see [19, 126, 33, 125], is based on finite sets as description methods. (The definitions can be extended to allow computable functions or probability distributions, instead of finite sets with essentially no effect on the resulting properties [33, 125].) An object x can be described using a finite set S that includes x, by sorting the elements of S in a prespecified order and specifying the index of x. The index can be encoded with a uniform code over $\{1, \ldots, |S|\}$, so that the ideal code-length equals $\log |S|$ bits. The code-length $\log |S|$ is actually also a lower bound that cannot be significantly beaten except for a very small subset of S, as can be seen by counting arguments, see e.g. [66]. A finite set S is called a *(Kolmogorov) sufficient statistic* for object x if we have[1]:

$$K(S) + \log |S| \leq K(x) + \mathcal{O}(1) \ . \tag{6.3}$$

In fact, the definition depends on the constant hidden in the $\mathcal{O}(1)$ notation. Such constants are usually ignored in the theory of Kolmogorov complexity; the following results hold for any 'large enough' value of the hidden constant.

The requirement (6.3) implies that x is *typical*, or random, as an element of S in the Martin-Löf sense [72]. To illustrate the idea of the sufficient statistic, consider the following simple properties. The singleton set $\{x\}$ is a sufficient statistic for all x, since $K(\{x\}) = K(x) + \mathcal{O}(1)$ and $\log |\{x\}| = 0$. In contrast, the set of all strings of length l_x is sufficient only for the uninteresting random strings with complexity $K(x) \geq l_x + K(l_x) + \mathcal{O}(1)$ with

[1] It is important to define exactly what is meant by $K(S)$. Namely, it must be required that $K(S)$ is the length of the shortest program that enumerates the elements of S *and then halts*. Otherwise, the set $S^k := \{y \ : \ K(y) \leq k\}$ becomes a sufficient statistic for every x with $K(x) = k$, and hence, all the regular features of x are summarized by stating that it belongs to the set of strings of complexity at most $K(x)$ and nothing more [33, Corollary III.13]. In the probabilistic version, this is equivalent to observing that all strings are random with respect to the universal distribution P_U [125]. Clearly, this fails to summarize the regular features in the data.

no regularity, except perhaps in their length. Since the two-part description based on S can never be shorter than the shortest description (not necessarily two-part), the inequality (6.3) can only hold as an equality (up to a constant).

The *Kolmogorov minimal sufficient statistic* (KMSS) is defined as the sufficient statistic of least complexity:

$$\mathrm{KMSS}(x) := \arg\min_{S} \{K(S) \ : \ K(S) + \log|S| = K(x) + \mathcal{O}(1)\} \quad .$$

The idea is to capture the regular (or "meaningful") information in x, leaving all the irregular or random features to be modeled as noise. To phrase this in terms of the aforementioned three concepts, we have the decomposition:

$$\begin{array}{rcccccc} K(x) & = & K(S) & + & \log|S| & + & \mathcal{O}(1) \\ \textit{complexity} & \approx & \textit{information} & + & \textit{noise}, & & \end{array}$$

Given data x, selecting the hypothesis (either a finite set or a probability distribution) that corresponds to the KMSS can be called 'ideal MDL'[2] [126].

It is also possible to consider the whole range of optimal statistics under the complexity restriction $K(S) \leq \alpha$ with α ranging between zero and $K(x)$; the behavior of such statistics is described by the *Kolmogorov structure function*, see [19, 124].

In practical MDL, the KMSS idea is implemented in a computable and non-asymptotic fashion. The code-length function $\mathcal{L}(D \ ; \ M_k)$ is known as the *stochastic complexity* of data D under model M_k. Its meaning is analogous to Kolmogorov complexity, the difference being that the set of all prefix Turing machines is replaced by the model M_k, and the universal Turing machine is replaced by a universal model. To define what is meant by a universal model, let the *regret* of distribution q for sequence x^n be defined as

$$\mathrm{REG}(x^n, q, M_k) := -\ln q(x^n) - \Big(-\ln p(x^n \mid \hat{\theta}_k(x^n), M_k)\Big) \quad , \qquad (6.4)$$

i.e., the excess code-length obtained when using q to encode x^n compared to what would have been the optimum achievable by model M_k, had the maximum likelihood parameters been known beforehand. A *universal model*

[2]Strictly speaking, the definition of ideal MDL by Vitányi and Li is slightly different from ours, but coincides with the KMSS decomposition under certain additional assumptions [126].

with respect to model M_k is a sequence of distributions, $p^1, p^2, \ldots$ on $\mathcal{X}^1, \mathcal{X}^2, \ldots$, such that the asymptotic per-symbol regret vanishes for all sequences $x_1, x_2, \ldots$ of increasing length (see [42, 43]):

$$\lim_{n\to\infty} \frac{1}{n}\mathrm{REG}(x^n, p^n, M_k) = 0 \ . \tag{6.5}$$

Hence, a universal model is able to imitate any distribution in the model M_k in the mean code-length sense up to some lower-order (sublinear) terms.

6.2 Stochastic complexity

There are three main types of universal codes used to define the stochastic complexity. Historically, the first one is based on *two-part codes*, where one first encodes optimally quantized parameter values, and then the data given the quantized parameters. For countable models it is not even necessary to use quantization to achieve universality. The second definition uses Bayes mixtures of the form (4.2), but without their Bayesian interpretation[3] [100]. Finally, the most recent definition of stochastic complexity is based on the Normalized Maximum Likelihood (NML) distribution [102], originally proposed by Shtarkov for data compression [116]. For sequences $x^n \in \mathcal{X}^n$, the NML distribution is defined as

$$p^n_{\mathrm{nml}}(x^n \ ; \ M_k) = \frac{p(x^n \mid \hat{\theta}_k(x^n), M_k)}{C^n_k} \ , \tag{6.6}$$

where the mapping $\hat{\theta}(\cdot)$ gives the maximum likelihood parameters, and the normalizing constant C^n_k is given in the discrete case by

$$C^n_k = \sum_{x^n \in \mathcal{X}^n} p(x^n \mid \hat{\theta}_k(x^n), M_k) \ , \tag{6.7}$$

and in the continuous case by the corresponding integral. From here on we will only discuss the discrete case, although all the results hold virtually unchanged when probability mass functions and sums are replaced by density functions and integrals.

[3] Paraphrased from [5]:

> " In the MDL principle for statistical inference there is no need for the awkward Bayesian interpretations of the meaning of the prior probability on the parameters. Rather, we may interpret distributions, such as $[Prob(D)]$, just as convex linear combinations of the models in the class, whose utility will be assessed on other grounds... " [103]

Encoding Continuous Outcomes: It is often said that when an outcome x^n is encoded using a density q, the code-length equals $-\ln q(x^n)$ nats. This is of course strictly speaking incorrect, since it is not possible to encode outcomes from an uncountable set with infinite precision using a countable set of code-words, i.e., some code-words should be infinitely long. In fact, it is commonly understood that continuous outcomes are discretized with some precision δ, high enough so that the probability mass of each quantization region is well approximated by $p(x^n) \cdot \delta^n$. This holds if the density is almost constant within a hyper-rectangle with side-length δ centered at x^n. The quantized outcomes can then be encoded with code-length approximately $-\ln p(x^n) - n \ln \delta$. Since the latter term is independent of the density, it is usually omitted.

The stochastic complexity based on the NML distribution becomes then

$$\mathcal{L}_{\text{nml}}(x^n \,;\, M_k) = -\ln p(x^n \mid \hat{\theta}_k(x^n), M_k) + \ln C_k^n \ .$$

The term *parametric complexity* is sometimes used for $\ln C_k^n$ since it gives the additional code-length incurred because the best parameter value $\hat{\theta}_k(D)$ is not known in advance. In some cases the parametric complexity is infinite, i.e., the normalizing integral diverges, which precludes the use of the NML universal distribution. This can be remedied by restricting the range of integration, using an altogether different universal code, or a combination of these, see [102, 109] and Papers 5 & 6 of this Thesis.

The NML universal distribution is optimal in the sense of the following two theorems.

Theorem 6.1 (Individual sequence minimax [116]) *When defined, the NML model is the unique solution to the minimax problem*

$$\inf_q \sup_{x^n \in \mathcal{X}^n} \text{REG}(x^n, q, M_k) \ , \tag{6.8}$$

where q can be any discrete distribution.

Theorem 6.2 (Expected minimax/maximin [104, 106]) *When defined, the NML model is the unique solution to the minimax problem*

$$\inf_q \sup_g \mathbb{E}_{x^n \sim g} \text{REG}(x^n, q, M_k) \ , \tag{6.9}$$

and the maximin problem

$$\sup_g \inf_q \mathbb{E}_{x^n \sim g} \text{REG}(x^n, q, M_k) \ , \tag{6.10}$$

where q and g can be any discrete distributions on sequences of length n.

Proof (of Theorems 6.1 & 6.2): Suppress the model M_k from notation for clarity. The first theorem follows directly from the observation that the NML model is the unique *equalizer strategy* with constant regret

$$\mathrm{REG}(x^n, p^n_{\mathrm{nml}}(\cdot)) \equiv \ln C^n_k \ ,$$

and that any other distribution must assign smaller probability than p^n_{nml} to at least one sequence x^n, and therefore, incurs greater loss for some x^n. The first part of the second theorem is actually almost identical to this, since we always have $\sup_g \mathbb{E}_{x^n \sim g} \mathrm{REG}(x^n, q) = \sup_{x^n \in \mathcal{X}^n} \mathrm{REG}(x^n, q)$. The second part follows by noticing that by definitions (6.6) and (6.4) we have

$$\begin{aligned} \mathrm{REG}(x^n, q) &= -\ln q(x^n) - (-\ln p_{\mathrm{nml}}(x^n)) + \ln C^n_k \\ &= -\ln \frac{q(x^n)}{g(x^n)} - \left(-\ln \frac{p_{\mathrm{nml}}(x^n)}{g(x^n)} \right) + \ln C^n_k \ , \end{aligned}$$

which implies the identity

$$\sup_g \inf_q \mathbb{E}_{x^n \sim g} \mathrm{REG}(x^n, q) = \sup_g \inf_q \mathrm{KL}(g \mid\mid q) - \mathrm{KL}(g \mid\mid p^n_{\mathrm{nml}}(\cdot)) + \ln C^n_k \ .$$

The theorem now follows, since $\mathrm{KL}(\cdot \mid\mid \cdot) \geq 0$ with equality if and only if the arguments coincide. The minmax value $\ln C^n_k$ is achieved by setting $g = q = p^n_{\mathrm{nml}}(\cdot)$. □

Both theorems hold also in the continuous case. Rissanen [104, 106] states Thm. 6.2 in terms of density functions, with an additional restriction on g which excludes singular distributions. However, if the theorem is formulated in terms of general probability measures and the associated concept of *divergence* (see e.g. [37, Ch. 5]), the above proof works for all distributions.

For parametric models satisfying suitable regularity assumptions, the stochastic complexity can be approximated analytically.

Theorem 6.3 (NML approximation [102]) *Under regularity conditions, the stochastic complexity under a k-parameter model M_k is approximated by*

$$\begin{aligned} \mathcal{L}(D \ ; \ M_k) = &-\ln p(D \mid \hat{\theta}_k(D), M_k) + \frac{k}{2} \ln \frac{n}{2\pi} \\ &+ \ln \int_\Theta \sqrt{\det I(\theta_k)} \, d\theta + o(1) \ , \end{aligned} \tag{6.11}$$

where $I(\cdot)$ is the Fisher information matrix, and the remainder term $o(1)$ goes to zero as $n \to \infty$.

There are different sets of regularity conditions that imply the theorem, see e.g. [102, 43]. The difference between the asymptotic expansions of stochastic complexity (6.11) and the Bayesian evidence (4.6) results from the prior-related terms. In fact, under similar regularity conditions as before, we can define Jeffreys' prior [55, 7]:

$$p_{\text{Jeffreys}}(\theta) := \frac{\sqrt{\det I(\theta)}}{\int_\Theta \sqrt{\det I(\eta)}\, d\eta} \ ,$$

which, when plugged into (4.6), results in identical asymptotic expansions. Like all approximations, this too should be used with care, see [87].

Rademacher complexity vs. parametric complexity: It is well-known that parametric complexity and the Bayesian evidence with Jeffreys' prior are closely related. It is also interesting to compare the parametric complexity $\ln C_k^n$ to Rademacher complexity (1.6). Both quantities measure how well the model is able to fit random data. To emphasize this similarity, we can rewrite (6.7) as

$$\sum_{x^n \in \mathcal{X}^n} p(x^n \mid \hat{\theta}_k(x^n), M_k) = |\mathcal{X}|^n \, \mathbb{E}_{x^n \sim \text{Uni}(\mathcal{X}^n)} \left[\sup_{\theta \in M_k} p(x^n \mid \theta, M_k) \right] \ .$$

It is easy to see, for instance, that the three intuitive properties of Rademacher complexity on p. 9 hold also for parametric complexity as defined using the NML universal model. In this sense it can be asserted that the correspondence between MDL and the SRM principle (p. 9) is more than superficial. However, it is hard to say if something could be achieved by analyzing these two in a common framework[4].

6.3 Prediction and model selection by MDL

The two main settings in which MDL is applied are prediction and model selection. We will only briefly mention some of the main issues in this direction. The literature on this topic is extensive, see e.g. [3, 43] and references therein.

6.3.1 Prediction

Consider first sequential prediction of outcomes $x_1, x_2, \ldots$, where the tth outcome is predicted based on the $t-1$ first outcomes. To simplify matters,

[4] Such an analysis is attempted in [121, Ch. 4] from a somewhat biased point of view.

assume that $q^1, q^2, \ldots$ is a universal model that constitutes a stochastic process, i.e., for all $t > 0$ and $x_1, \ldots, x_{t-1}$ we have

$$\sum_{x_t \in \mathcal{X}} q^t(x_1, \ldots, x_t) = q^{t-1}(x_1, \ldots, x_{t-1}) \ .$$

By Kolmogorov's extension theorem we can now let q denote the (unique) distribution over infinite sequences from which a sequence of finite-length distributions, $q^1, q^2, \ldots$ can be obtained. For instance, the Bayes mixture universal model is a stochastic process, while the NML universal model is not. What is achieved by the restriction to stochastic processes is that it is now straightforward to consider the asymptotic behavior of the sequence of predictions $(q(x_t \mid x^{t-1}))_{t=1}^{\infty}$.

The problem of prediction is closely related to compression: if the predictions are probability distributions over outcomes, $q(x_t \mid x^{t-1})$, and loss is measured by log-loss, then the loss is actually given by the code-length, and *vice versa*. It is therefore immediate that universal models satisfying (6.5) are good predictors in the sense that the cumulative regret (excess log-loss) with respect to the best element in the reference class grows at most sublinearly in n. Furthermore, under a Gaussian model, log-loss is determined by the squared errors, and thus, compression can also be identified with regression estimation (p. 4).

In order to relate compression to something more familiar from a (frequentist) statistical point of view, we can assume that the data are generated by a distribution in model M. We can then consider whether the MDL predictor is also *consistent* in the sense that its risk (expected loss) converges to the minimum achievable under the given model M. Adhering to log-loss, the risk is given by the expected code-length, i.e., entropy, of the conditional distribution $q(\cdot \mid x^{t-1})$. If data is generated by distribution p^*, the minimum of this is achieved by $p^*(\cdot \mid x^{t-1})$. We can now consider the excess risk incurred by q, given by the Kullback-Leibler divergence $\mathrm{KL}(p^*(\cdot \mid x^{t-1}) \mid\mid q(\cdot \mid x^{t-1}))$.

It turns out that all universal models are consistent in terms of so called *Cesàro consistency*, but not necessarily in terms of the standard notion of consistency.

Definition 6.1 (Cesàro consistency) *Given a stochastic process q, the expected KL risk at step t under distribution p^* is given by*

$$\mathcal{E}^t_{\mathrm{KL}}(p^*, q) := \mathbb{E}_{x^{t-1} \sim p^*} \mathrm{KL}(p^*(\cdot \mid x^{t-1}) \mid\mid q(\cdot \mid x^{t-1})) \ ,$$

where the expectation is over the initial sequence x^{t-1}; and the Cesàro KL

risk is given by

$$\mathcal{E}^t_{\mathrm{CKL}}(p^*, q) := \frac{1}{t} \sum_{i=1}^{t} \mathcal{E}^t_{\mathrm{KL}}(p^*, q) \ .$$

We call p KL consistent *if, for all $p^* \in M$, the KL risk vanishes as $t \to \infty$, and similarly,* Cesàro consistent *if, for all $p^* \in M$, the Cesàro risk vanishes as $t \to \infty$.*

Theorem 6.4 (Cesàro consistency [2]) *The predictions of a universal model are Cesàro consistent, but not necessarily KL consistent.*

The difference is that KL consistency requires that for all $\epsilon > 0$, the risk eventually becomes smaller than ϵ and *never goes up again*, while Cesàro consistency allows that the risk may exceed ϵ for arbitrarily large t, as long as this occurs less and less frequently. For further discussion, see [43].

6.3.2 Model selection

The original and still predominant application of MDL is model selection, see [99, 34, 63, 42, 86]. In order to measure performance, we can consider a nested set of models $M_1 \subset M_2 \subset \ldots$, and assume that the data are generated by a distribution p^* which is an element of at least one of the models. We denote by M^* the smallest model that includes p^* — since the models are nested M^* is well-defined. A model selection method is called consistent if, loosely speaking, it eventually finds the model M^*. It can be shown that, under regularity conditions, model selection by the MDL principle (6.1) is consistent, see [3, 43].

While the regularity conditions necessary to prove consistency of MDL model selection are too technical to be stated here, it should be emphasized that they are by no means automatically satisfied. In fact, there are certain 'pathologic' cases where the conditions are violated and MDL overfits, i.e., chooses too complex a model, and continues to do so even with increasing sample-size. Perhaps the most striking example of this is the Csiszár-Shields anomaly [21]: when estimating the order of a Markov chain from pure random data, i.e., data generated by a Bernoulli model with parameter exactly 1/2, the estimated order grows unboundedly with increasing sample size. The problem does not occur if the Bernoulli parameter of the generating distribution differs from 1/2, or if the singleton model Bernoulli(1/2) is included in the set of allowed models. In Chapter 8 we encounter a similar phenomenon.

6.4 Discussion

The foundations of MDL have some features that sets it apart from most other frameworks for data analysis. One of these is the departure from the assumption that there is a 'true' data-generating distribution. This is related to the attitude of the probability-theorist de Finetti who stated that "probability does not exist" [30]. What de Finetti meant was, however, that probability does not exist as an *objective* phenomenon, but that it does exist in the subjectivistic Bayesian sense, and that subjective probability can and should be used as a basis for making decisions. In MDL, the data are not assumed to be generated by a distribution, nor it is assumed that subjective degrees of belief have any bearing on valid statistical inferences.

In the preceding sections, we explained the rationale of the MDL principle in terms of Kolmogorov's minimal sufficient statistic decomposition, and its non-asymptotic embodiment as 'practical MDL' of Rissanen. In addition to such a justification from first principles, so to speak, *if* we make some assumptions on the data, then it can be shown that MDL methods work, although there are some subtle issues related to the exact way in which performance is measured, and even some (arguably unrealistic) cases where MDL methods can fail.

Since the MDL principle is designed to extract information from data, it is sometimes unclear how — or even whether — it should be applied in decision-theoretic problems where a specific loss function is considered. For instance, if the loss function is not log-loss, good compression does not ensure good predictive performance. This is similar to the generative vs. discriminative aspect of Bayesian theory. To overcome this difficulty, variants of MDL methods that are tailored for specific loss functions have been suggested [134, 41, 107]. These touch upon the so called 'expert framework' [127, 13]. The expert framework is a variant of the statistical learning framework, similar in spirit to MDL in the sense that no assumptions are made about the data-generating mechanism, and that performance is measured in terms of worst-case *relative* loss. For instance, the extended stochastic complexity of Yamanishi [134] coincides with Vovk's Aggregating Algorithm [127] from the expert framework in terms of the predictions they yield. Thus, MDL and the expert setting complement each other in a way that lends support to both of them.

Chapter 7

Compression-Based Stemmatic Analysis

Before the development of the art of printing, pioneered by Gutenberg in the 15th century, written works were copied by hand. This resulted in numerous unintentional errors that accumulated in copies of copies, copies of copies of copies, etc. Consequently, a text of any importance ended up existing in a group of different variants, some of them all but identical to the original, some perhaps hardly recognizable. Connecting each variant to its *exemplar* (the variant from which it was copied), gives a tree-like structure called the *stemma*, with the original version as the root. The aim of stemmatology is to recover this structure given a set of surviving variants.

There is an obvious analogy in evolutionary biology to the transmission of textual information in the stemma. Namely, the transmission of genetic information and the development of species, often visualized as a *phylogenetic tree* or, more poetically, the 'Tree of Life'[1], has the same characteristics of unintentional errors and iterative multiplication as 'manuscript evolution'. The methods developed for phylogenetic analysis have been fruitfully adapted and applied to stemmatology, see e.g. [108, 119].

In Paper 4, we present a method for stemmatic analysis. The core of the method is a compression-based criterion for comparing stemmata.

7.1 An MDL criterion

One of the most applied methods in phylogenetics is maximum parsimony. A maximally parsimonious tree minimizes the total number of differences

[1] See `http://www.tolweb.org`.

between connected nodes — i.e., species, individuals, or manuscripts that are directly related — possibly weighted by their importance. In stemmatology the analysis is based on variable readings that result from unintentional errors in copying or intentional omissions, insertions, or other modifications. Our MDL criterion shares many properties of the maximum parsimony method. In line with the MDL principle, we measure the total description length of all the variants corresponding to a given stemma, and choose the stemma that minimizes the code-length.

Intuitively, the idea in the MDL criterion is the following. All variants are described by picking one of them as a starting point, proceeding along the edges of the stemma tree to the tips of the branches, or the *leafs*, and describing each variant along they way given its already described predecessor. Having described the predecessor of a variant, the new variant can be described concisely if it resembles the predecessor. Hence, a stemma where similar variants are placed in neighboring nodes gives a shorter code-length than a stemma where similar variants are randomly scattered across different branches.

In order to define the code-length of a string given another string we need to choose a specific code. The universal Kolmogorov complexity (see Sec. 6.1) is noncomputable, and defined only up to a constant which may be significant for short strings. In the spirit of a number of earlier authors (see [40, 17, 129] and references therein), we approximate Kolmogorov complexity by using a compression program (`gzip`). We also modify the `gzip` complexity by letting the complexity $C(x \mid x)$ be zero for all x, and ignoring certain features known to be uninformative[2].

Formally, the total code-length given a graph G is computed by first picking a root node and considering the directed version of G where each edge is directed away from the root, towards the leafs. Given such a directed graph $\vec{G}$, the code-length is given by

$$C(\vec{G}) = \sum_{v \in V(\vec{G})} C(v \mid \mathrm{pa}(v, \vec{G})) = \sum_{v \in V(\vec{G})} C(\mathrm{pa}(v, \vec{G}), v) - C(\mathrm{pa}(v), \vec{G}) \ , \tag{7.1}$$

where $V(\vec{G})$ is the set of nodes (vertices) of the graph, and $\mathrm{pa}(v, \vec{G})$ denotes the parent of node v in $\vec{G}$. If node v has no parent, $\mathrm{pa}(v, \vec{G})$ is defined as the empty string.

For simplicity, and following the common practice in phylogenetics, we

[2]Ignoring uninformative features was achieved by removing the differences between the variants with respect to such features. For instance, all occurrences of the ampersand '&' were replaced by the word *et*, and all occurrences of the letter v were replaced by the letter u.

restrict the stemma to a bifurcating tree, i.e., a tree in which all interior nodes have exactly three neighbors. Since in any realistic case, some of the manuscripts are missing, it is not reasonable to build a stemma consisting only of the surviving manuscripts. Instead, the remaining variants are all placed in the leaf nodes of the stemma, and the interior nodes are reserved for the missing variants. Note that even though some of the interior nodes may actually be available among the set of remaining variants, we can always imagine that those variants are duplicated so that the original text is lost and the copy is placed in a leaf node. Missing leaf nodes, i.e., missing variants with no surviving descendants have no practical significance. If the code-length of a pair, $C(x, y)$, is symmetric in the sense $C(x, y) = C(y, x)$, which is approximately true in our application, the right-hand side of (7.1) becomes for all bifurcating trees

$$C(G) = \sum_{(v,w)\in E(G)} C(v, w) - 2 \sum_{v\in V_I(G)} C(v) \ ,$$

where $E(G)$ denotes the set of edges in G, and $V_I(G)$ denotes the set of *interior* nodes in G. Hence the choice of the root node is irrelevant. In other words, the method gives no indication of the temporal order in the stemma.

7.2 Optimization algorithms

From an algorithmic point of view, the task of finding both a tree structure and the contents of the missing nodes is a daunting combinatorial optimization problem. Fortunately, given a tree structure, the optimal interior node contents minimizing the total code-length can be found in polynomial time in the number of nodes, under certain restrictions. More specifically, we compute the cost $C(v \mid \mathrm{pa}(v, \vec{G}))$ in (7.1) as a sum of the contributions of segments of 10–20 consecutive words, and assume that the possible choices for the contents of each segment in the interior nodes are those appearing in the segment in question in at least one of the available variants[3]. To simplify notation, consider a fixed (directed) graph, and a fixed segment. Let the different versions of the segment in the available variants be denoted by $x^1, \ldots, x^m$. Under the restriction that $x^1, \ldots, x^m$ are the only possible choices, the minimum achievable code-length per the segment, and given

[3]For instance, if the available variants are $(AACB, ABCB, BBAB, BBBA)$, where A, B, C are used in place of the segments, then the possible interior nodes are $AAAA, AAAB, AABA, AABB, \ldots, BBCB$. This requires that the variants are aligned so that each segment corresponds to the same part of the text in all variants.

the graph, can be evaluated using a dynamic programming solution with the recursion at the interior nodes (see [29]):

$$\mathrm{cost}_i(j) = \min_k \Big[C(x^k \mid x^j) + \mathrm{cost}_a(k) \Big] + \min_l \Big[C(x^l \mid x^j) + \mathrm{cost}_a(l) \Big] \ ,$$

where a and b are the children of node i. The recursion is initialized at the leaf nodes by letting

$$\mathrm{cost}_i(j) = \begin{cases} 0, & \text{if } x^j \text{ matches the content of node } i; \\ \infty, & \text{otherwise.} \end{cases}$$

The total cost of the tree is obtained by summing over the segments the minimal costs

$$\min_j \mathrm{cost}_{\mathrm{root}}(j) + C(x^j) \ .$$

Assuming that computing the code-length $C(x^k \mid x^j)$ can be done in constant time for all k and j, the time-complexity of the algorithm is of order $\mathcal{O}(knm^2)$, where n is the number of nodes, k is the number of segments, and m is the maximum number of different versions of a segment. In the worst case, all the versions of all segments differ, in which case we have $m = n$, and the time-complexity is of order $\Theta(kn^3)$.

With respect to the tree structure, the situation is not as easy. The number of different bifurcating trees is superexponential. Hence exhaustive search is infeasible, and no feasible alternative guaranteed to find the optimal tree is known. We use simulated annealing [56], accepting random modifications to the tree with probability

$$p := \min \left\{ 1, \exp\left(\frac{\text{total-cost}_{\mathrm{old}} - \text{total-cost}_{\mathrm{new}}}{T} \right) \right\} \ ,$$

where T is a temperature parameter that is slowly decreased to zero. When evaluating the total cost, the algorithm also takes advantage of the fact that small modifications require only partial updating of the dynamic programming tables. With a large enough initial choice of T, the initialization of the tree has no practical significance. We ran several runs up to 2.5 million iterations, each of which usually resulted in a very similar final tree structure and total cost.

7.3 Results and future work

Figure 7.1 and Table 7.1 illustrate the method by a simple example with five variants, each consisting of five words. The segment length is set to one

	x		y	$C(y \mid x)$
1.	sanctus	→	beatus	5
2.	ex	→	in	3
3.	henricus	→	Henricus	3
4.	Anglia	→	anglia	3
5.	ex	→	in	3

Table 7.1: Conditional complexity of the modifications relevant to the example in Figure 7.1, as obtained from the `gzip` compressor. The complexity $C(x \mid x)$ is forced to be zero for all x.

word. In the main experiment, we analyzed all the known 52 variants of the Legend of St. Henry of Finland [49]. The obtained tree is largely supported by more traditional analysis in earlier work, and points out groups of related manuscripts not discovered before. For more details, see Paper 5.

We are currently carrying out controlled experiments with artificial (hand-copied) data with known 'ground-truth' solution to which the results can be compared[4]. Outside historical and biological applications, analysis of computer viruses is an interesting research topic, see [129]. As further research topics, it would be interesting to investigate ways to overcome some of the restrictions of the method. Most importantly it would be more realistic *not* to restrict to bifurcating trees — in reality, manuscripts were sometimes copied from multiple exemplars, manifesting as non-treelike structures. Currently, such generalizations are mostly unexplored in both stemmatology and phylogenetics.

[4]See `http://www.cs.helsinki.fi/teemu.roos/casc/`.

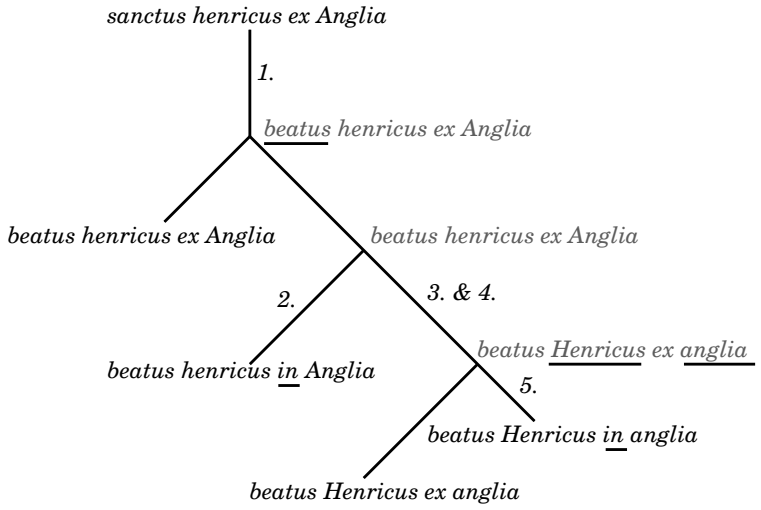

Figure 7.1: An example tree obtained with the compression-based method for the five strings at the tips of the branches. Changes are underlined and numbered. Costs of changes are listed in Table 7.1 using the same numbering as in the graph. Best reconstructions at interior nodes are shown at the branching points. The solution is not unique.

Figure 7.2: An excerpt of a 15th century manuscript 'H' from the collections of the Helsinki University Library, showing the beginning of the legend of St. Henry on the right: *"Incipit legenda de sancto Henrico pontifice et martyre; lectio prima; Regnante illustrissimo rege sancto Erico, in Suecia, uenerabilis pontifex beatus Henricus, de Anglia oriundus, ..."* [49].

Chapter 8

MDL Denoising

Denoising means the process of removing noise from a signal. This may be necessary due to an imprecise measurement device or transmission over a noisy channel. Traditional techniques, such as mean and median filters that operate directly on the signal, remove in effect the high-frequency components from the signal. This often removes a large fraction of the noise, but in some cases leads to loss of too much detail. They also require that some parameters such as window size, etc., are tuned, usually by hand, to find a suitable balance between noise reduction and resolution.

Time-frequency transforms, including wavelet transforms, enable better resolution by operating both in the frequency domain and the time (spatial) domain, see [70]. A hierarchy of denoising methods is presented in Fig. 8.1.

As explained in Chapter 6, the MDL principle is by its very purpose designed to separate information and noise, and hence naturally applicable to denoising. In Papers 5 & 6, we analyze an extend an MDL denoising method of Rissanen [105]. The developed methods are freely available at the author's web-page[1].

8.1 Wavelet regression

We focus on the regression-type case where the signal is a sequence of real-valued measurements, $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ (for convenience transposed to get a column vector). Two-dimensional signals are represented in the same sequential form by reading the measurements in a row-by-row or column-by-column order. Let $\mathcal{W}$ be an $n \times m$ regressor matrix (the choice of the letter $\mathcal{W}$ becomes clear shortly) whose columns give the basis vectors $\{(w_{1,j}, \ldots, w_{n,j})\}_{j=1}^{m}$. The standard linear regression model (see Chapter 2)

[1] `http://www.cs.helsinki.fi/teemu.roos/denoise/`

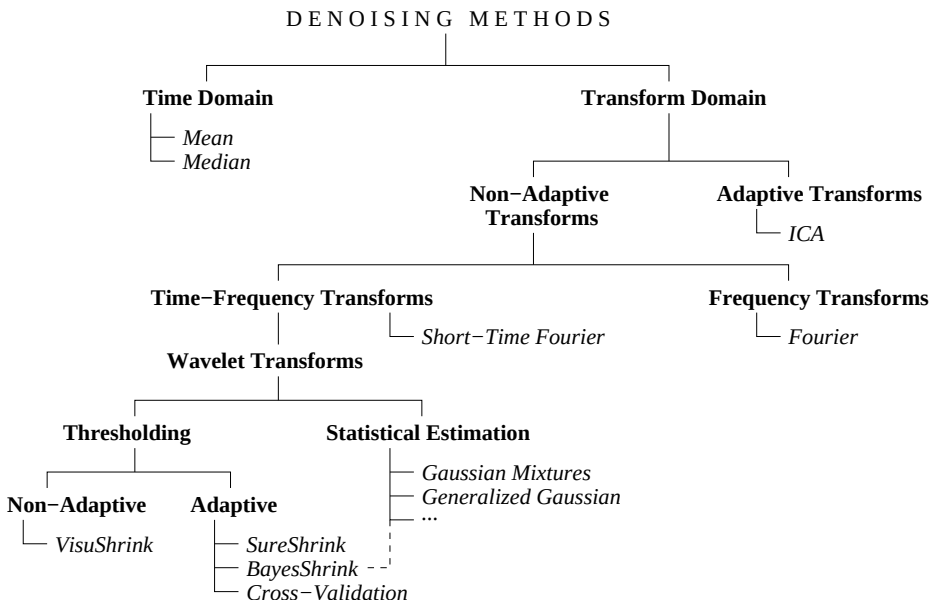

Figure 8.1: A partial hierarchy of denoising methods with emphasis on wavelet-based approaches (adapted from [84]): The methods are grouped into ones that operate directly on the signal (time domain) and ones that apply transformations (transform domain). Wavelet-based methods are further grouped according to the type of operations performed on the wavelet coefficients. (The groups are not mutually exclusive: for instance, in BayesShrink the optimal threshold value is determined using statistical estimation under the generalized Gaussian model.)

gives the observed signal $\mathbf{y}$ as a linear combination of the basis vector plus noise:

$$\mathbf{y} = \mathcal{W}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_N^2) \ , \tag{8.1}$$

where the noise sequence $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^{\mathrm{T}}$ is taken to be i.i.d. Gaussian with variance, σ_N^2.

Using an orthonormal wavelet basis as the regressor matrix $\mathcal{W}$ (which explains the letter $\mathcal{W}$, for wavelet) implies that the basis vectors are orthogonal unit vectors. This restriction is satisfied by, for instance, the Haar basis and Daubechies family of bases, see [70]. Orthonormality has several computational and statistical advantages [69]. One of the main computational advantages is that, by the identity $\mathcal{W}^{\mathrm{T}}\mathcal{W} = \mathbf{I}$, the least-squares solution (recall Eq. (2.2)) simplifies:

$$\hat{\boldsymbol{\beta}} = (\mathcal{W}^{\mathrm{T}}\mathcal{W})^{-1}\,\mathcal{W}^{\mathrm{T}}\mathbf{y} = \mathcal{W}^{\mathrm{T}}\mathbf{y} \ ,$$

and that furthermore, there is a fast (linear-time) algorithm for the evaluation of $\mathcal{W}^{\mathrm{T}}\mathbf{y}$, known as the Fast Wavelet Transform (FWT), similar to

the Fast Fourier Transform (FFT). The statistical advantages are related to the fact that most natural signals have sparse wavelet representations — the distribution of wavelet coefficients is heavy-tailed — while Gaussian i.i.d. noise is unaffected by the transform.

The idea of wavelet thresholding is to apply a parameterized thresholding function to the wavelet coefficients. In the simplest form, known as *hard thresholding*, the thresholding function sets all coefficients whose absolute value is below a threshold, T, to zero, and leaves the remaining ones intact. In another popular choice, called *soft thresholding*, the procedure is otherwise the same as in hard thresholding, except that the threshold parameter T is also subtracted from the absolute values of the remaining coefficients. There are various approaches to choosing the value of the thresholding parameter, each giving rise to a different denoising method, e.g., VisuShrink and SureShrink [24]; and BayesShrink [14] (see Fig. 8.1).

8.2 Codes and models for wavelet coefficients

For complete wavelet bases with $m = n$ basis vectors, the maximum likelihood (i.e., least-squares) fit gives $\mathbf{y} = \mathcal{W}\hat{\boldsymbol{\beta}}$, leaving nothing to be modeled as noise. Hard thresholding can be considered as choosing a subset, γ, of the basis vectors, and projecting the signal orthogonally to the space spanned by the chosen vectors via $\mathcal{W}_\gamma \mathcal{W}_\gamma^{\mathrm{T}} \mathbf{y}$, where $\mathcal{W}_\gamma$ denotes the reduced matrix comprising only of the basis vectors γ. The critical question is then: which one of the subsets should be chosen?

As mentioned above, hard thresholding has been studied both in the frequentist and Bayesian frameworks. Rissanen [105] suggests to choose the basis vectors by the MDL principle[2]. In order to define the length of the description of the observed signal, he uses a special two-fold NML (or *re*normalized maximum likelihood, RNML) procedure.

8.2.1 Renormalized NML

In the first phase of the RNML procedure, the free parameters to be maximized in the NML model are the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$, and the noise variance σ_N^2. However, the data have to be restricted by *hyper-parameters*, or otherwise the normalizing coefficient giving the parametric complexity becomes infinite. To nullify the effect of the restriction on the criterion, a

[2]Regarding the hierarchy of Figure 8.1, MDL denoising methods are most naturally placed in the "Statistical Estimation" branch, since in them thresholding is more a consequence than a design solution, the primary aim being model selection.

second-level NML model is constructed by treating the hyper-parameter as the free parameters to be maximized. For any subset of the basis vector indices, $\gamma \subseteq \{1, \ldots, n\}$, of cardinality $k = |\gamma|$, the RNML code-length is well approximated by[3]

$$\mathcal{L}_{\text{rnml}}(\mathbf{y}\ ;\ \gamma) \approx \frac{n-k}{2} \ln \frac{S(\mathbf{y}) - S_\gamma(\mathbf{y})}{n-k} + \frac{k}{2} \ln \frac{S_\gamma(\mathbf{y})}{k} + \frac{1}{2} \ln k(n-k) + C \ , \tag{8.2}$$

where $S_\gamma(\mathbf{y}) = \sum_{i \in \gamma} \hat{\boldsymbol{\beta}}_i^2$, $S(\mathbf{y}) := S_{\{1,\ldots,n\}}(\mathbf{y})$, and the additive constant C is independent of γ and $\mathbf{y}$. The only approximation step is the Stirling approximation of the Gamma function, which is very accurate. The criterion is always minimized choosing in γ some k smallest or largest coefficients in absolute value [105], which allows huge computational savings compared to trying all the 2^n subsets. It seems that in most practical situations — and in fact, we argue in Papers 5 & 6 that this holds in *all* situations — the largest coefficients in absolute value should be retained.

8.2.2 An equivalent NML model

Since the renormalization procedure is not as well understood as the standard NML model, it is useful to know that the code-length function (8.2) can be obtained using the standard NML under a slightly different model. The new model includes a density for the $\boldsymbol{\beta}$ coefficients, for which reason we call it the 'extended' model. The extended model is given by

$$\mathbf{y} = \mathcal{W}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_N^2), \quad \begin{cases} \beta_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \tau^2), & \text{if } i \in \gamma, \\ \beta_i = 0, & \text{otherwise.} \end{cases} \tag{8.3}$$

A similar model is often used in Bayesian variable selection, where it is dubbed the *spike-and-slab* model [82]. The spike-and-slab model corresponds to the sparseness property: the spike produces a lot of coefficients near zero (but in practice not exactly zero due to noise), while the slab gives the heavy tails.

In Paper 5 it was hinted that the NML code constructed from such an extended model by integrating over the $\boldsymbol{\beta}$ coefficients and maximizing with respect to τ and σ, agrees with the RNML code (8.2) constructed from the standard regression model (8.1). This claim is proved in Paper 6. The advantage of such an alternative derivation of the same criterion is in the insight it gives to the procedure. For instance, the overfitting problem occurring in the high-noise regime identified in Paper 5 can be traced to

[3] In [105] the third term $(1/2) \ln k(n-k)$ was incorrectly in the form $(1/2) \ln k/(n-k)$.

the fact that fitting two Gaussian densities to data from a single Gaussian density gives nonsensical results. Even more importantly, once the underlying model is well understood, it can be easily modified and generalized in a meaningful way.

8.3 Three refinements

It is customary to ignore the encoding of the index of the model class in MDL model selection (see Eq. (6.1)). One simply picks the class that enables the shortest description of the data without considering how many bits are needed to indicate which class was used. However, when the number of different model classes is large, like in denoising where it is 2^n, the code-length for the model index can not be omitted.

Encoding a subset of k indices from the set $\{1, \ldots, n\}$ can be done very simply by using a uniform code over the $\binom{n}{k}$ subsets of size k. This requires that the number k is encoded first, but this part can be ignored if a uniform code is used, which is possible since the maximum n is fixed. Adding the code-length of the model index to the code-length of $\mathbf{y}$ given γ, Eq. (8.2), gives the total code-length

$$\mathcal{L}(\gamma) + \mathcal{L}(\mathbf{y}\ ;\ \gamma) \approx \frac{n-k}{2} \ln \frac{S(\mathbf{y}) - S_\gamma(\mathbf{y})}{(n-k)^3} + \frac{k}{2} \ln \frac{S_\gamma(\mathbf{y})}{k^3} + C', \qquad (8.4)$$

where C' is a constant independent of γ, and the only approximative step is again the Stirling approximation, which is very accurate. This gives refinement A to Rissanen's [105] MDL denoising method.

It is well-known that in natural signals, especially images, the distribution of the wavelet coefficients is not constant across the so called *subbands* of the transformation. Different subbands correspond to different orientations (horizontal, vertical, diagonal), and different scales. Letting the coefficient variance, τ^2, depend on the subband produces a variant of the extended model (8.3). The NML code for this variant can be constructed using the same technique as for the extended model with only one adjustable variance. The resulting code-length function becomes after the Stirling approximation as follows:

$$\sum_{b=0}^{B} \left(\frac{k_b}{2} \ln \frac{S_{\gamma^b}(\mathbf{y})}{k_b} + \frac{1}{2} \ln k_b \right) + \sum_{b=1}^{B} \ln \binom{n_b}{k_b} + C'' \ , \qquad (8.5)$$

where B is the number of subbands, γ^b denotes the set of retained coefficients in subband b, $k_b := |\gamma^b|$ denotes their number, n_b denotes the total number of coefficients in subband b, and C'' is constant with respect to γ.

Algorithm 1 Subband adaptive MDL denoising

Input: Signal y^n.
Output: Denoised signal.

1: $c^n \leftarrow \mathcal{W}^T y^n$
2: **for all** $b \in \{1, \ldots, B\}$ **do**
3: $\quad k_b \leftarrow n_b$
4: **end for**
5: **repeat**
6: $\quad$ **for all** $b \in \{B_0 + 1, \ldots, B\}$ **do**
7: $\quad\quad$ optimize k_b wrt. criterion (8.5)
8: $\quad$ **end for**
9: **until** convergence
10: **for all** $i \in \{1, \ldots, n\}$ **do**
11: $\quad$ **if** $i \notin \gamma$ **then**
12: $\quad\quad c_n \leftarrow 0$
13: $\quad$ **end if**
14: **end for**
15: **return** $\mathcal{W} c^n$

Finding the coefficients that minimize criterion (8.5) simultaneously for all subbands can no longer be done as easily as previously. In practice, a good enough solution is found by an iterative optimization of each subband while letting the other subbands be kept in their current state, see Algorithm 1. In order to make sure that the coarse structure of the signal is preserved, the coarsest B_0 subbands are not processed in the loop of Steps 5–9. In the condition of Step 11, the final model γ is defined by the largest k_b coefficients on each subband b. This gives refinement B.

Refinement C is inspired by predictive universal coding with weighted mixtures of the Bayes type, used earlier in combination of mixtures of trees [130]. The idea is to use a mixture of the form

$$p_{\mathrm{mix}}(\mathbf{y}) := \sum_{\gamma} p_{\mathrm{nml}}(\mathbf{y} \; ; \; \gamma) \pi(\gamma) \;\; ,$$

where the sum is over all the subsets γ, and $\pi(\gamma)$ is the prior distribution corresponding to the $\ln \binom{n}{k}$ code defined above. This is similar to Bayesian model averaging (4.5) except that the model for $\mathbf{y}$ given γ is obtained using NML. This induces an 'NML posterior', a normalized product of the prior and the NML density. The normalization presents a technical difficulty since in principle it requires summing over all the 2^n subsets. In Paper 6, we present a computationally feasible approximation which turns out to

lead to a general form of soft thresholding. The soft thresholding variation can be implemented by replacing Step 12 of Algorithm 1 by the instruction

$$c_i \leftarrow c_i \frac{\tilde{r}_i}{1 + \tilde{r}_i} \ ,$$

where $\tilde{r}_i$ is a ratio of two NML posteriors which can be evaluated without having to find the normalization constant.

All three refinements improve the performance, measured in terms of peak-signal-to-noise ratio or, equivalently, mean squared error, in the artificial setting where a 'noiseless' signal is contaminated with Gaussian noise, and the denoised signal is compared to the original. Figures 8.2 and 8.3 illustrate the denoising performance of the MDL methods and three other methods (VisuShrink, SureShrink [24], and BayesShrink [14]) for the Doppler signal [24] and the Barbara image[4]. The used wavelet transform was Daubechies D6 in both cases. In terms of PSNR, the refinements improve performance in all cases except for one: refinement A decreases PSNR for the Barbara image, Fig. 8.3. For more results, see Paper 6, and the supplementary material[5].

The best method in the Doppler case is the MDL method with all three refinements, labeled "MDL (A-B-C)" in the figures. For the Barbara image, the best method is BayesShrink. The difference in the preferred method between the 1D signal and the image is most likely due to the fact that the generalized Gaussian model used in BayesShrink is especially apt for natural images. However, actually none of the compared methods are currently state-of-the-art for image denoising, where the best special-purpose methods are based on overcomplete (non-orthogonal) wavelet decompositions, and take advantage of inter-coefficient dependencies, see e.g. [93]. Applying the MDL approach to special-purpose image models is a future research goal. In 1D signals such as Doppler, where the new method has an advantage, it is likely to be directly useful.

[4] From `http://decsai.ugr.es/~javier/denoise/`.

[5] All the results in Paper 6 (and some more), together with all source code, are available at `http://www.cs.helsinki.fi/teemu.roos/denoise/`.

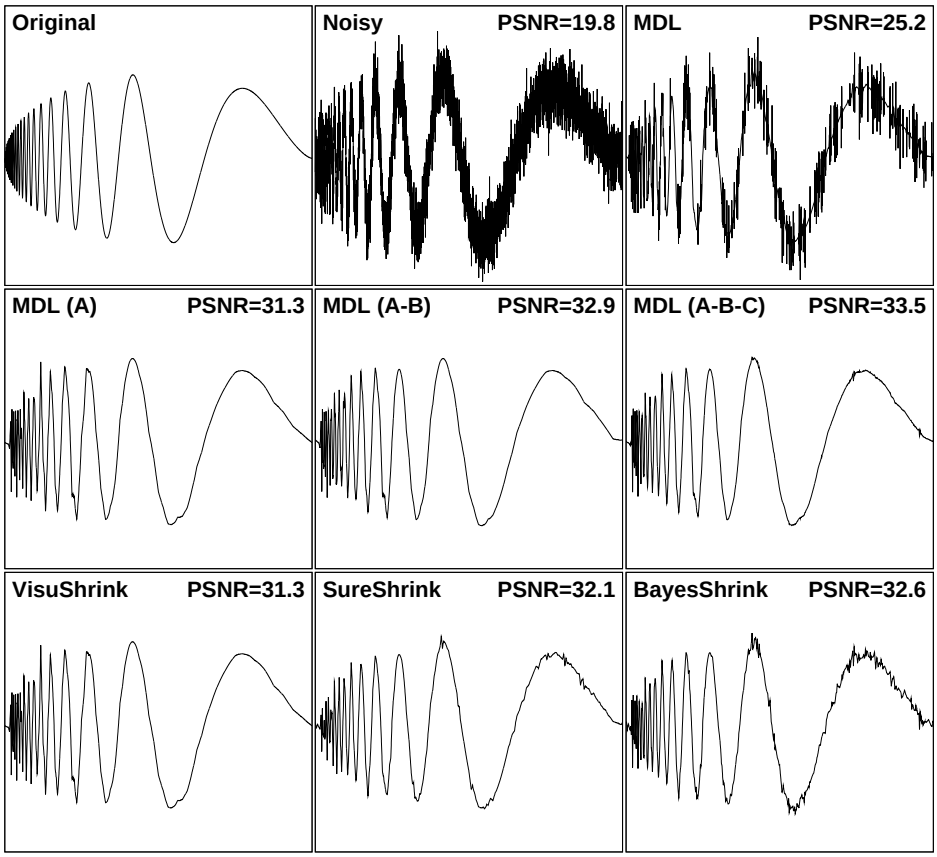

Figure 8.2: Doppler signal [24]. *First row:* original signal, sample size $n = 4096$; noisy signal, noise standard deviation $\sigma = 0.1$; original MDL method [105]. *Second row:* MDL with refinement A; MDL with refinements A and B; MDL with refinements A, B, and C. *Third row:* VisuShrink; SureShrink; BayesShrink. Peak-signal-to-noise ratio (PSNR) in decibels is given in each panel. (Higher PSNR is better). The denoised signals of MDL (A) and VisuShrink are identical (PSNR=31.3 dB).

Figure 8.3: Barbara image (detail). *First row:* original image; noisy image, noise standard deviation $\sigma = 20.0$; original MDL method [105]. *Second row:* MDL with refinement A; MDL with refinements A and B; MDL with refinements A, B, and C. *Third row:* VisuShrink; SureShrink; BayesShrink. Peak-signal-to-noise ratio (PSNR) in decibels is given in each panel. (Higher PSNR is better).

References

[1] Ole Barndorff-Nielsen. *Information and Exponential Families.* John Wiley & Sons, New York, NY, 1978.

[2] Andrew R. Barron. Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 6, pages 27–52. Oxford University Press, 1998.

[3] Andrew R. Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[4] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[5] Rohan A. Baxter and Jonathan J. Oliver. MDL and MML: Similarities and differences (Introduction to minimum encoding inference — Part III). Technical Report 207, Department of Computer Science, Monash University, Clayton, Vic., 1994.

[6] James O. Berger. *Statistical Decision Theory: Foundations, Concepts, and Methods.* Springer-Verlag, New York, NY, 1980.

[7] José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory.* John Wiley & Sons, New York, NY, 1994.

[8] David Blackwell and Lester Dubins. Merging of opinions with increasing information. *Annals of Mathematical Statistics*, 33(3):882–886, 1962.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[10] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning: ML Summer Schools 2003*, volume 3176 of *Lecture Notes in Artificial Intelligence*, pages 169–207. Springer-Verlag, Heidelberg, 2004.

[11] Wray Buntine. Theory refinement on Bayesian networks. In B. D'Ambrosio and P. Smets, editors, *Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann, 1991.

[12] Bradley P. Carlin and Siddhartha Chib. Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B*, 57(3):473–484, 1995.

[13] Nicoló Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[14] S. Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

[15] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[16] Siddhartha Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

[17] Rudi Cilibrasi and Paul M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[18] Bertrand S. Clarke and Andrew R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, 36(3):453–471, 1990.

[19] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991.

[20] Richard T. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.

[21] Imre Csiszár and Paul C. Shields. The consistency of the BIC Markov order estimator. *Annals of Statistics*, 28(6):1601–1619, 2000.

[22] A. Philip Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

[23] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with Discussion). *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[24] David L. Donoho and Iain M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[25] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1st edition, 1973.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork. *Patter Classification*. John Wiley & Sons, New York, NY, 2nd edition, 2000.

[27] Ian R. Dunsmore. Asymptotic prediction analysis. *Biometrika*, 63(3):627–630, 1976.

[28] William Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, New York, NY, 3rd edition, 1968.

[29] Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA, 2004.

[30] Bruno de Finetti. La prévision: Ses lois logiques, ses sources subjectives. *Annales de l'Institute Henri Poincaré*, 7:1–68, 1937. Reprinted as 'Foresight: Its logical laws, its subjective sources' in H. E. Kyburg and H. E. Smokler, editors, *Studies in Subjective Probability*, Dover, 1964.

[31] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997.

[32] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3/4):601–620, 2000.

[33] Péter Gács, John T. Tromp, and Paul M. B. Vitányi. Algorithmic statistics. *IEEE Transactions on Information Theory*, 47(6):2443–2463, 2001.

[34] Qiong Gao, Ming Li, and Paul M. B. Vitányi. Applying MDL to learn best model granularity. *Artificial Intelligence*, 121(1–2):1–29, 2000.

[35] Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, 1996.

[36] Irving J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3–4):237–264, 1953.

[37] Robert M. Gray. *Entropy and Information Theory*. Springer-Verlag, New York, NY, 1990.

[38] Russell Greiner, Adam J. Grove, and Dale Schuurmans. Learning Bayesian nets that perform well. In D. Geiger and P. P. Shenoy, editors, *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence*, pages 198–207. Morgan Kaufmann, 1997.

[39] Daniel Grossman and Pedro Domingos. Learning Bayesian network classifiers by maximizing conditional likelihood. In C. E. Brodley, editor, *Proceedings of the 21st International Conference on Machine Learning*, pages 361–368. ACM Press, 2004.

[40] Stéphane Grumbach and Fariza Tahi. A new challenge for compression algorithms: Genetic sequences. *Journal of Information Processing and Management*, 30(6):875–866, 1994.

[41] Peter D. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, University of Amsterdam, The Netherlands, 1998.

[42] Peter D. Grünwald. A Tutorial introduction to the minimum description length principle. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in MDL: Theory and Applications*. MIT Press, Cambridge, MA, 2005.

[43] Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007. Forthcoming.

[44] Peter D. Grünwald, Petri Kontkanen, Petri Myllymäki, Teemu Roos, and Henry Tirri. Supervised posterior distributions. Presented at the 7th Valencia Meeting on Bayesian Statistics, Tenerife, Spain, 2002.

[45] Joseph Y. Halpern. Cox's theorem revisited (Technical addendum). *Journal of Artificial Intelligence Research*, 11:429–435, 1999.

[46] Mark H. Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[47] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[48] David Heckerman and Christopher Meek. Embedded Bayesian network classifiers. Technical Report MSR-TR-97-06, Microsoft Research, Redmond, WA, 1997.

[49] Tuomas Heikkilä. *Pyhän Henrikin Legenda* (in Finnish). Suomalaisen Kirjallisuuden Seura, Helsinki, Finland, 2005.

[50] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[51] Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: A tutorial (with Discussion). *Statistical Science*, 14(4):382–417, 1999.

[52] Colin Howson and Peter Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court, La Salle, IL, 1989.

[53] Edwin T. Jaynes and G. Larry Bretthorst. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, MA, 2003.

[54] Tony Jebara. *Machine Learning: Discriminative and Generative*. Kluwer, Boston, MA, 2003.

[55] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Journal of the Royal Statistical Society. Series A*, 186(1007):453–461, 1946.

[56] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[57] Andrey N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[58] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[59] Petri Kontkanen, Petri Myllymäki, Tomi Silander, and Henry Tirri. On supervised selection of Bayesian networks. In K. Laskey and H. Prade, editors, *Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence*, pages 334–342. Morgan Kaufmann, 1999.

[60] Leon G. Kraft. *A Device for Quantizing, Grouping, and Coding Amplitude-Modulated Pulses*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1949.

[61] Lawrence Krauss. *Quintessence: The Mystery of Missing Mass in the Universe*. Basic Books, New York, NY, 2000.

[62] John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

[63] Aaron D. Lanterman. Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model selection. *International Statistical Review*, 69(2):185–212, 2001.

[64] Julia A. Lasserre, Christopher M. Bishop, and Thomas P. Minka. Principled hybrids of generative and discriminative models. In A. Fitzgibbon, Y. LeCun, and C. J. Taylor, editors, *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 87–94. IEEE Computer Society, 2006.

[65] Steffen L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, UK, 1996.

[66] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1993.

[67] Dennis Lindley. *Making Decisions*. John Wiley & Sons, New York, NY, 2nd edition, 1985.

[68] David J. C. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, Pasadena, CA, 1991.

[69] Stéphane Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[70] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, 1998.

[71] M. E. Maron. Automatic indexing: An experimental inquiry. *Journal of the ACM*, 8:404–417, 1961.

[72] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9(6):602–619, 1966.

[73] David McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 164–170. ACM Press, 1999.

[74] David McAllester and Luiz E. Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4:895–911, 2003.

[75] David McAllester and Robert E. Schapire. On the convergence rate of Good-Turing estimators. In S. A. Goldman N. Cesa-Bianchi, editor, *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pages 1–6. Morgan Kaufmann, 2000.

[76] David McAllester and Robert E. Schapire. Learning theory and language modeling. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, pages 271–287. Morgan Kaufmann, San Francisco, CA, 2003.

[77] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, NY, 1997.

[78] Brockway McMillan. Two inequalities implied by unique decipherability. *IRE Transactions on Information Theory*, 2(4):115–116, 1956.

[79] Thomas P. Minka. Algorithms for maximum-likelihood logistic regression. Technical Report 758, Department of Statistics, Carnegie Mellon University, Pittsburth, PA, 2001. Revised Sept. 2003.

[80] Thomas P. Minka. Discriminative models, not discriminative training. Technical Report MSR-TR-2005-144, Microsoft Research, Cambridge, UK, 2005.

[81] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[82] Toby J. Mitchell and John J. Beauchamp. Bayesian variable selection in linear regression (with discussion). *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

[83] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, 1997.

[84] Mukesh C. Motwani, Mukesh C. Gadiya, Rakhi C. Motwani, and Frederick C. Harris, Jr. Survey of image denoising techniques. In *Proceedings of the Global Signal Processing Expo and Conference*, 2004.

[85] Iain Murray and Zoubin Ghahramani. A note on the evidence and Bayesian Occam's razor. Technical report, Gatsby Computational Neuroscience Unit, University College London, 2005.

[86] In Jae Myung, Daniel J. Navarro, and Mark A. Pitt. Model selection by normalized maximum likelihood. *Journal of Mathematical Psychology*, 50(2):167–179, 2006.

[87] Daniel J. Navarro. A note on the applied use of MDL approximations. *Neural Computation*, 16(9):1763–1768, 2004.

[88] David Newman, Seth Hettich, Catherine Blake, and Christopher Merz. UCI repository of machine learning databases. University of California, Irvine, CA, 1998.

[89] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression on naive Bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 605–610. MIT Press, 2001.

[90] Jonathan J. Oliver and Rohan A. Baxter. MML and Bayesianism: Similarities and differences (Introduction to minimum encoding inference — Part II). Technical report, Department of Computer Science, Monash University, Clayton, Vic., 1994.

[91] Alon Orlitsky, Narayana P. Santhanam, and Junan Zhang. Always Good Turing: Asymptotically optimal probability estimation. In M. Sudan, editor, *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 179–188. IEEE Computer Society, 2003. Also: *Science*, 302(5644):427–431, 2003.

[92] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[93] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of Gaussians

in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

[94] Frank P. Ramsey. Truth and probability. In R. B. Braithwaite, editor, *The Foundations of Mathematics and other Logical Essays*, chapter VII, pages 156–198. Kegan, Paul, Trench, Trubner & Co., London, 1931.

[95] Theodore S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, Upper Saddle River, USA, 1996.

[96] Carl E. Rasmussen and Zoubin Ghahramani. Occam's razor. In T. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 294–300. MIT Press, 2000.

[97] Gunnar Rätsch. *Robust Boosting via Convex Optimization: Theory and Applications*. PhD thesis, University of Potsdam, Germany, 2001.

[98] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[99] Jorma Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14(3):1080–1100, 1986.

[100] Jorma Rissanen. Stochastic complexity (with discussion). *Journal of the Royal Statistical Society. Series B*, 49(3):223–239, 253–265, 1987.

[101] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, New Jersey, 1989.

[102] Jorma Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996.

[103] Jorma Rissanen. Information theory and neural nets. In P. Smolensky, M. C. Mozer, and D. E. Rumelhart, editors, *Mathematical Perspectives on Neural Networks*. Lawrence Erlbaum Associates, 1996.

[104] Jorma Rissanen. A generalized minmax bound for universal coding. In *Proceedings of the 2000 IEEE International Symposium on Information Theory*, page 324. IEEE Press, 2000.

[105] Jorma Rissanen. MDL denoising. *IEEE Transactions on Information Theory*, 46(7):2537–2543, 2000.

[106] Jorma Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Transactions on Information Theory*, 47(5):1712–1717, 2001.

[107] Jorma Rissanen. Complexity of simple nonlogarithmic loss functions. *IEEE Transactions on Information Theory*, 49(2):476–484, 2003.

[108] Peter M. W. Robinson and Robert J. O'Hara. Report on the Textual Criticism Challenge 1991. *Bryn Mawr Classical Review*, 3(4):331–337, 1992.

[109] Steven de Rooij and Peter Grünwald. An empirical study of minimum description length model selection with infinite parametric complexity. *Journal of Mathematical Psychology*, 50(2):180–192, 2006.

[110] Y. Dan Rubinstein and Trevor Hastie. Discriminative vs informative learning. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 49–53. AAAI Press, 1997.

[111] Leonard J. Savage. *The Foundations of Statistics*. John Wiley & Sons, New York, NY, 1954.

[112] Cullen Schaffer. A conservation law for generalization performance. In W. Cohen and H. Hirsh, editors, *Proceedings of the 11th International Conference on Machine Learning*, pages 259–265. Morgan Kaufmann, 1994.

[113] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[114] Loraine Schwartz. On consistency of Bayes procedures. *Proceedings of the National Academy of Sciences*, 52(1):46–49, 1964.

[115] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[116] Yuri M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):175–186, 1987.

[117] Adrian F. M. Smith and David J. Spiegelhalter. Bayes factors and choice criteria for linear models. *Journal of the Royal Statistical Society. Series B*, 42(2):213–220, 1980.

[118] Ray J. Solomonoff. A formal theory of inductive inference, Parts 1 & 2. *Information and Control*, 7:1–22, 224–254, 1964.

[119] Matthew Spencer, Klaus Wachtel, and Christoper J. Howe. The Greek Vorlage of the Syra Harclensis: A comparative study on method in exploring textual genealogy. *TC: A Journal of Biblical Textual Criticism* [http://purl.org/TC], 7, 2002.

[120] William Talbott. Bayesian epistemology. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2006 edition, 2006.

[121] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.

[122] Vladimir N. Vapnik. An overview of statistical learing theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[123] Vladimir N. Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[124] Nikolai K. Vereshchagin and Paul M. B. Vitányi. Kolmogorov's structure functions and model selection. *IEEE Transactions on Information Theory*, 50(12):3265–3290, 2004.

[125] Paul M. B. Vitányi. Meaningful information. *IEEE Transactions on Information Theory*, 52(10):4617–4626, 2006.

[126] Paul M. B. Vitányi and Ming Li. Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Transactions on Information Theory*, 46(2):446–464, 2000.

[127] Vladimir Vovk. Aggregating strategies. In M. A. Fulk and J. Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–386. Morgan Kaufmann, 1990.

[128] Chris S. Wallace and David M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, 1968.

[129] Stephanie Wehner. Analyzing worms and network traffic using compression. *Journal of Computer Security*, 15(3):303–320, 2007.

[130] Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995.

[131] David H. Wolpert. On the connection between in-sample testing and generalization error. *Complex Systems*, 6(1):47–94, 1992.

[132] David H. Wolpert. The lack of *a priori* distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 1996.

[133] David H. Wolpert. The supervised learning no-free-lunch theorems. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, and F. Hoffmann, editors, *Soft Computing and Industry: Recent Applications*, pages 25–42. Springer-Verlag, 2002.

[134] Kenji Yamanishi. A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Transactions on Information Theory*, 44(8):1424–1439, 1998.

[135] Alexander K. Zvonkin and Leonid A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.