17 *MDL in Context*

In this chapter, we compare refined MDL to various other statistical inference methods. We start in Section 17.1, with a comparison between MDL and various frequentist approaches. Section 17.2 considers the relation between MDL and Bayesian inference in great detail. Section 17.3 compares MDL model selection to the two popular default model selection approaches AIC and BIC. Section 17.4 compares MDL to the similar Minimum *Message* Length Principle. Section 17.5 compares MDL to Dawid's prequential approach. Sections 17.6 through 17.10 consider cross-validation, maximum entropy, idealized MDL, individual sequence prediction and statistical learning theory, respectively. The latter two comparisons make clear that there do exist some problems with the MDL approach at its current stage of development. These are discussed in Section 17.11, in which I also suggest how the MDL approach might or should develop in the future.

Unless mentioned otherwise, whenever we write "MDL," we refer to the individual-sequence version of MDL; see the introduction to Part III of this book.

A Word of Warning Some of the differences between MDL and other approaches simply concern details of algorithms that are used for practical inference tasks. Others concern the difference in underlying principles. Discussion of such philosophical differences is inherently subjective, and therefore, some comparisons (especially MDL-Bayes, Section 17.2) will be quite opinionated. When referring to the underlying principles, I use phrases such as "From an MDL perspective, . . .," "the MDL view," and so on. In those cases where I strongly agree with the MDL perspective, I have permitted myself to use first-person phrases such as "in my opinion." I have tried to restrict the use of phrases involving the word "we" (such as "we now see that") to "ob-

jective" facts, such as mathematical derivations. These safeguards notwithstanding, it is inevitable that in this chapter, my personal viewpoints become entangled with those of Rissanen and other MDL researchers.

17.1 MDL and Frequentist Paradigms

Against Principles? Frequentist or "orthodox" statisticians (Chapter 2, page 73) are often suspicious of anything calling itself a "principle."

> I have heard several well-known statisticians say this. One well-known statistical learning theorist even told me that "MDL and Bayes are just *recipes* for doing inferences; these may work in some, but not in other situations. In contrast, in learning theory we design algorithms that are provably optimal."

They argue that, when designing methods for learning, rather than dogmatically adhering to principles, we should focus on what counts: making sure that our algorithms learn a good approximation to the true distribution fast, based on as few data as possible. Researchers who dismiss principles such as MDL on such grounds tend to forget that they themselves strongly rely on another principle, which one might call *the frequentist principle*: the idea that it is realistic to view real-world data as sampled from some P^* in some model class $\mathcal{M}$. This is just as much a "principle" as the basic MDL idea that "the more you can compress data, the more you have learned from the data." From an MDL perspective, there are at least three problems with the frequentist principle:

1. **Too unrealistic.** The frequentist principle is often quite unrealistic in practice. In Section 17.1.1 we illustrate this using a detailed example, and we emphasize that frequentist ideas can be used either as a sanity check for algorithms based on other principles, or as a principle for *designing* learning algorithms. From the MDL stance, only the latter type of frequentist principle is truly problematic.

2. **Too restrictive.** The frequentist design principle is unnecessarily restrictive. The reason is that frequentist procedures usually violate the *prequential principle*, which we discuss in Section 17.1.2.

3. **Too much work.** Designing optimal frequentist procedures involves a lot of work, which may sometimes be unnecessary. This is explained in Section 17.1.3.

In the sections below we discuss each issue in turn. Before we start, we note that the first issue (unrealistic probabilistic assumptions) is mostly resolved in *statistical learning theory*, which we further discuss in Section 17.10. This is a frequentist approach to learning from data that only makes very mild assumptions about the data generating distribution. The price to pay is that it can be applied only in restricted settings.

17.1.1 Frequentist Analysis: Sanity Check or Design Principle?

Frequentist Assumptions Are Unrealistic We already argued in Chapter 1, Section 1.7, that frequentist assumptions are often unrealistic. It is sometimes argued that the frequentist principle becomes more realistic if one takes a nonparametric approach. Thus, one assumes that data are sampled from some P^* in some large nonparametric class $\mathcal{M}^*$. For example, as in Chapter 13, Section 13.3, one could choose to "merely" assume that P^* has a differentiable density f^* on its support $\mathcal{X} = [0, 1]$, where $f^*(x)$ is bounded away from 0 and infinity. While this is evidently a weaker assumption than the parametric requirement that P^* is a member of some finite-dimensional model, it is still problematic from an MDL perspective. Namely, such nonparametric modeling is often applied in settings where it is unclear whether the assumption of a "true" P^* makes sense at all. And even if one does assume that such a P^* exists, the assumption that its density be differentiable is still very strong, and one would need an infinite amount of data to test whether it truly holds. Often, it will simply not hold, as Example 17.1 illustrates: data may be sampled from a distribution P^* which has no density, yet $\mathcal{M}^*$ may contain distributions with differentiable densities that predict data sampled from P^* very well. However, if we design our estimators so as to be optimal from a frequentist perspective, we may end up with a brittle algorithm that only works well (learns P^* fast) if P^* truly has a density, and fails dramatically if P^* has no density.

> For example, consider any learning algorithm for histogram density estimation that achieves the minimax optimal convergence Hellinger rate (Chapter 13, Chapter 16). We can always modify such an algorithm such that, if for some $n, n' > n$, $x_n = x_{n'}$, then for *all* $n'' > n'$, the algorithm outputs the uniform distribution on $\mathcal{X}$. Hence the algorithm breaks down as soon as two times the same outcome is observed. Since such an event has probability 0 under any $P^* \in \mathcal{M}^*$, it will still achieve the minimax optimal convergence rate.
>
> While most nonparametric estimators used in practice will not do anything strange like this, and will be reasonably robust against P^* that do not admit

a density, the point is that there is nothing in the frequentist paradigm which requires one to avoid brittle algorithms like the one above.

To me, it seems a much better strategy to design a learning method which leads to good predictions whenever some $P \in \mathcal{M}^*$ leads to good predictions, no matter what particular sequence is observed. By Theorem 15.1, the resulting algorithms then also lead to consistent estimates, such that *if* data are i.i.d. $\sim P^*$ according to some $P^* \in \mathcal{M}^*$, *then* the estimator converges to P^* at reasonably fast rate. This is a fantastic sanity check of such an individual-sequence method; but one should never turn the world on its head, and design estimators solely to converge fast if $P^* \in \mathcal{M}^*$; for then one cannot really say anything about their performance on the sequence that actually arrives.

Example 17.1 [Models with, and Distributions without Densities] One often adopts the normal family as one's statistical model because one has reason to believe that each outcome X_i is really the normalized sums of a number of bounded i.i.d. random variables $Y_{i,1}, \ldots, Y_{i,m}$. It then follows by the central limit theorem that the X_i are approximately normally distributed. However, if the $Y_{i,j}$ are discrete (for example, Bernoulli) random variables, then the X_i will have a discrete rather than a continuous distribution. In particular, the X_i will not have a differentiable density. Modeling such data by a normal distribution is harmless as long as one considers individual sequences: with high probability, $\hat{\theta}(x^n)$ will achieve small log loss $-\log f_{\hat{\theta}(x^n)}(x^n)$, and universal models relative to the normal family will lead to good predictions of future data in the log loss sense. Thus, even in a parametric context, a model $\mathcal{M}$ of distributions with densities can be a very good model for the data even though the "true" distribution does not have a density, but only if the estimators based on such models do not crucially rely on the existence of such a density. One would expect that the same is the case for nonparametric models.

Summarizing, from an MDL perspective, the frequentist principle may very well serve as a sanity check, but never as a design principle (an explicit example of the frequentist design principle at work is given in Section 17.10). To be fair, I should immediately add that many frequentist statisticians have been working in a way consistent with the "weakly frequentist" sanity-check view. They accept that some of the most clever statistical procedures in history have been suggested by external ideas or principles, and they make it their business to analyze the frequentist behavior of such procedures. This type of research, which may analyze methods such as moment-based estimators, Bayes procedures, cross-validation, MDL, and many others, is generally illuminating and often quite sophisticated. However, one surprising aspect

from an MDL point of view is that a substantial fraction of this research still concentrates on the maximum likelihood method and its direct extensions.[1]

The Strange Focus on Maximum Likelihood Fisher's ideas on maximum likelihood estimation provided an enormous breakthrough in the 1920s and 1930s. While at that time, there was ample justification to study its frequentist properties, I think that now, in 2006, it is time to move on. The ML method suffers from a number of problems: (1) it does not protect against overfitting; (2) in some quite simple problems (such as the Neyman-Scott problem (Wallace 2005)), its performance is simply dismal; (3) even if one analyzes its behavior for quite simple parametric families, it is sometimes not "admissible" (Ferguson 1967). Finally, (4), it provides no clue as to why there exist phenomena such as "superefficiency." A superefficient estimator relative to a parametric Θ is one which, like the ML estimator, achieves the minimax (squared Hellinger) convergence rate $O(n^{-1})$, but achieves much faster rates on a subset of Θ of Lebesgue measure 0. None of these four issues are a problem for MDL. Given this insight, it seems strange to focus so much effort on proving consistency and convergence rate results for ML estimators.

> For example, overfitting is taken care of automatically by MDL, witness the convergence results Theorem 15.1 and Theorem 15.3; its behavior on Neyman-Scott is just fine (Rissanen 1989); admissibility is guaranteed for two-part MDL estimators (Barron and Cover 1991), and finally, the superefficiency phenomenon is easily explained from an MDL point of view (Barron and Hengartner 1998): as we already pointed out in Chapter 6, one can easily design a Bayesian universal code relative to a parametric model Θ with a discrete prior, that puts nonzero prior mass on all rational-valued parameter values. One can define a meta-universal code relative to this code, combined with the ordinary Bayesian universal code based on a continuous prior. Such a meta-universal code will achieve standard expected redundancy $O(\log n)$ if $X_1, X_2, \ldots \sim P_\theta$ for all $\theta \in \Theta$, but it will achieve expected redundancy $O(1)$ for θ with rational-valued parameters. It follows that for rational parameters, the KL, and therefore, the Hellinger risk, of the corresponding prequential Bayesian estimator must essentially converge (Chapter 15, Corollary 15.1, page 473) at rate faster than $O(f(n))$, where $\sum_{i=1,2,\ldots} f(i) < \infty$, so that $f(n) = o(1/n)$.

1. To witness, I have once taken part in a discussion with several well-known non-Bayesian statisticians, who simply could not believe (a) that in the MDL approach to histogram estimation, histograms are determined by a Jeffreys' or Laplace estimator, rather than an ML estimator (Chapter 13, Section 13.3); and (b) that this matters.

ML Is Fundamental - in a Different Sense The ML estimator plays a fundamental role in statistics: it pops up in a wide variety contexts, it has a natural feel to it, and it has many pleasant properties. To give but one example of such a property, in exponential families, it is equal to the average sufficient statistic of the observed data. Because of its fundamental role, it is sometimes argued that there is ample justification to continue studying the convergence properties of ML estimators. From an MDL point of view, the ML estimator is indeed a most fundamental notion, so there is no contradiction here. But it is invariably seen as a quantity that is optimal *with hindsight*. Thus, the MDL goal is to design estimators or predictors that predict almost as well as the ML estimator constructed with hindsight from the data that needs to be predicted. As explained in Chapter 15, Section 15.4.1, this goal is *not* achieved by predicting with, or estimating by, the ML estimator itself.

Of course, one can extend the ML method to deal with complex models by adding complexity penalties, as is done, in, for example, the AIC model selection criterion (Section 17.3.2). Such complexity penalties are typically designed such as to achieve good rates of convergence in terms of (e.g., Hellinger) risk. Thus, they are designed so as to achieve good performance in expectation, where the expectation is under one of the distributions in the assumed model class. The problem with such extensions is that now the frequentist paradigm is once more used as a design principle, rather than merely as a sanity check.

17.1.2 The Weak Prequential Principle

Inference methods based on the frequentist design principle often violate the *weak prequential principle (WPP)*. The WPP was introduced in essence by Dawid (1984), and investigated in detail by Dawid and Vovk (1999).

Let us consider a hypothesis P for data $x_1, \ldots, x_n$ that can be used for sequentially predicting x_i given x^{i-1}. For example, P may be a probabilistic source, or a universal code representing a set of probabilistic sources. According to the WPP, the quality of P as an explanation for given data $x^n = x_1, \ldots, x_n$ should only depend on the actual outcomes $x_1, \ldots, x_n$, as well as the predictions that the distribution P makes on x^n, i.e. on the set of n conditional probabilities $P(x_i \mid x^{i-1})$, $i = 1, ...n$. It should not depend on any other aspect of P. In particular, the conditional probability of any $x \in \mathcal{X}$ conditioned on data y^{i-1} that did *not* occur in the sequence $x_1, \ldots, x_n$ should not play any role at all.

There are at least two reasons why this might be a good idea: first, intu-

itively, it would be strange or even irrational, if predictions *that were never made* would somehow influence any decision about whether P is a suitable model for the observed data. Second, we may sometimes want to consider a "hypothesis" P whose predictions conditional on unseen data are simply *unknowable*. A prototypical example is weather forecasting.

Example 17.2 [Weather Forecasters and the WPP] Here we let P represent a weather forecaster, in the following sense: let data $(x_1, y_1), \ldots, (x_n, y_n)$ represent consecutive days. On each day $i-1$, based on previous data x^{i-1}, y^{i-1}, the weather forecaster announces the probability that it will rain on day i. Thus, $y_i \in \mathcal{Y}$ indicates whether it rains (1) or not (0) on day i, and the weather forecaster's predictions can be thought of as conditional distributions $P(Y_i \mid x^{i-1}, y^{i-1})$. Here each x_i can be thought of as a gigantic vector summarizing all the observable data on day i that the forecaster makes use of in her prediction algorithm. This may include air pressure, humidity, temperature and other related quantities measured at various places around the world.

In the Netherlands we have two weather forecasters (one working for public television, the other for commercial television). Both make daily predictions about the precipitation probability for the next day. If we want to know who of the two we should listen to, we would like to compare their predictions on some sequence of days, say, the entire previous year. If we use a comparison procedure which needs to look at their prediction for day y_i in contexts that have not taken place (i.e. for values of x^{i-1} and y^{i-1} different from those observed), then we are in trouble, for it would be exceedingly hard to obtain this information.

The WPP is violated by many classical statistical procedures, including standard null hypothesis testing. As a result, standard hypothesis testing cannot be used to determine the quality of a weather forecaster, merely by watching her make predictions on television. Instead, one would need to know what she *would* have predicted in all possible situations that might have, but did not occur. The relation between the WPP and MDL is explored in Section 17.5.

17.1.3 MDL vs. Frequentist Principles: Remaining Issues

As claimed by the frequentist in the beginning of this section, page 524, principles like MDL and Bayes do provide recipes to "automatically" approach all kinds of statistical problems. But unlike the frequentist, I think this is good rather than bad. The alternative offered by the frequentist design principle is to design separate, possibly entirely different algorithms for each of the many

different types of inductive inference tasks, such as classification, regression, sequential prediction, model selection, clustering, similarity analysis... For each of these tasks, one should design an algorithm with good properties for exactly that task. To me, such an approach seems neither very elegant nor very robust. I'd much rather use a "principle" (such as MDL or Bayes) that is widely applicable, always yields reasonable answers, even if in any particular application, the method induced by the principle is not 100% optimal. For example, it is not clear to me whether one should be particularly concerned about the fact that the risk of MDL inference with CUP model classes in nonparametric contexts does not always converge at the optimal rate (Chapter 16, Section 16.6). MDL sometimes needs an extra $\log n$ factor in expectation compared to the minimax optimal algorithm, under some assumptions on the true distribution P^*. This may not be a very large price to pay, given that we have designed our MDL algorithm without making any assumptions about this P^* whatsoever!

However, the fact that I (and most other individual-sequence MDL adherents) embrace the use of frequentist analysis as a sanity check, does imply the following: suppose that an MDL procedure has *really bad* frequentist behavior, e.g. suppose that it would be inconsistent under a wide variety of conditions. Then I am in trouble: my basic principle suggested a method which fails my sanity check. Luckily, the convergence results Theorem 15.1 and 15.3 guarantee that this never happens when the model class contains the true data-generating distribution. However, such inconsistency can sometimes occur if the model class is misspecified; see Section 17.10.2.

Expectation-Based vs. Individual-Sequence MDL As discussed in Chapter 6, Section 6.5, a majority of information theorists works with stochastic rather than individual-sequence universal codes, where usually, "universality" is defined in expectation with respect to some P in one of the models under consideration. According to the individual-sequence MDL philosophy, one really should not use expectation-based MDL procedures, which are based on such expected-redundancy universal models. Nevertheless, I have to admit that from a certain point of view, the use of such codes is quite natural. To see this, first note that if we code data using the code corresponding to a distribution P, then the code would be optimal in expectation sense if indeed the data were distributed $\sim P$. That is, we associate each distribution P with the code Q achieving

$$\min_Q D(P\|Q),$$

where Q ranges over all codelength functions. The Q achieving this minimum happens to be equal to P. Starting from that consideration, if we want to associate a code with a *set* of distributions $\mathcal{M}$, the natural extension seems to be to require that the code would be optimal in expectation sense if indeed the data were distributed $\sim P$, in the worst-case over all $P \in \mathcal{M}$. Thus, we should pick the code minimizing

$$\min_{Q} \max_{P \in \mathcal{M}} D(P\|Q).$$

This is exactly what we do if we base model selection on the minimax optimal expected redundancy code ((6.41) on page 202) rather than the minimax optimal individual-sequence regret code $\bar{P}_{\rm nml}$.

Thus, one may reason that the proper type of universal code to use in MDL inference is of the expected rather than the individual sequence kind. In my own personal view, expectation-based MDL is an interesting variation of MDL where the frequentist principle is elevated from its sanity check status, and put on the same footing as the compression principle. I prefer the individual-sequence principle, but think it is reassuring that in practice, individual-sequence MDL procedures are usually also minimax optimal in an expectation sense, and *intuitive*[2] expectation-based MDL procedures often also turn out to work well in an individual sequence sense; see, however, the discussion in Chapter 11, the discussion below Example 11.12, page 326. In the remainder of this chapter, the term "MDL" keeps referring to the individual-sequence version.

17.2 MDL and Bayesian Inference

Bayesian statistics is one of the most well-known, frequently and successfully applied paradigms of statistical inference (Berger 1985; Bernardo and Smith 1994; Lee 1997; Gelman et al. 2003). It is often claimed that "MDL is really just a special case of Bayes."[3] Although there are close similarities, this is simply not true. To see this quickly, consider the basic quantity in refined MDL: the NML distribution $\bar{P}_{\rm nml}$, (6.15) on page 181. There is no mention of anything like this code/distribution in any Bayesian textbook!

2. The word "intuitive" is meant to rule out brittle procedures such as those described on page 525 which, for example, crash if the same observation occurs twice.

3. The author has heard many people say this at many conferences. The reasons are probably historical: while the underlying philosophy has always been different, until Rissanen introduced the use of $\bar{P}_{\rm nml}$, most actual implementations of MDL "looked" quite Bayesian.

Thus, it must be the case that Bayes and MDL are somehow different. While a Bayesian statistician may still think of NML as an approximation to the log marginal likelihood (see below), this cannot be said for the "localized" NML distribution (14.14) on page 427. While natural from an MDL point of view, this version of NML cannot even be interpreted as an approximation to any Bayesian quantity. The differences become more dramatic if one considers expectation-based MDL as well. In Section 17.3.2 below we describe an expectation-based MDL method that combines the best of the AIC and BIC model selection criteria, and that does not seem to resemble any Bayesian procedure.

In the remainder of this section, we analyze the differences between MDL and Bayes in considerable detail. We first give a high-level overview, emphasizing the difference in underlying principles. Then, in Section 17.2.1 through 17.2.3, we investigate the practical consequences of these underlying differences of principle.

The MDL vs. the Bayesian Principles Two central tenets of modern Bayesian statistics are: (1) probability distributions are used to represent uncertainty, and to serve as a basis for making predictions, rather than merely standing for some imagined "true state of nature"; and, (2), all inference and decision-making is done in terms of prior and posterior distributions and utility functions. MDL sticks with (1) (although here the "distributions" are primarily interpreted as "codelength functions"), but not (2): MDL allows the use of arbitrary universal models such as NML and prequential plug-in universal models; the Bayesian universal model does not have a special status among these. Such codes are designed according to the minimum compression (or "maximum probability") principle and the luckiness principle, both of which have no direct counterpart in Bayesian statistics.

MDL's Two Principles: Maximum Probability and Luckiness The first central idea of MDL is to base inferences on universal codes that achieve small codelength in a minimax sense, relative to some class of candidate codes ("model") $\mathcal{M}$. This minimum codelength paradigm may be reinterpreted as a *maximum probability principle*, where the maximum is relative to some given models, in the worst case over all sequences (Rissanen (1987) uses the phrase "*global* maximum likelihood principle"). Thus, whenever the Bayesian universal model is used in an MDL application, a prior (usu-

ally Jeffreys') should be used that minimizes worst-case codelength regret, or equivalently, maximizes worst-case relative probability.

In practice, the minimax optimal prior is often not well defined. Then MDL approaches have to resort to a second idea: the luckiness principle. The procedure now becomes "subjective,"[4] just like a Bayesian approach. Still, there remain some essential differences between MDL and Bayes. The most important of these are:

1. **Types of universal codes.** MDL is not restricted to Bayesian universal codes; for example, LNML and plug-in codes may be used as well.

2. **Hope vs. expectation.** The luckiness-type subjectivity of MDL is of an inherently different, weaker type than the subjectivity in Bayesian approaches. This is explained in Section 17.2.1 below. As a consequence, many types of inferences and decisions that are sometimes made in Bayesian inference are meaningless from an MDL perspective. This is perhaps the most crucial, yet least understood difference between MDL and Bayes.

3. **Priors must compress.** Even when luckiness functions are allowed, if a Bayesian marginal likelihood is used in an MDL context, then it has to be interpretable as a universal code, i.e. it has to compress data. This rules out the use of certain priors. For example, the Diaconis-Freedman type priors, which make Bayesian inference inconsistent, cannot be used in MDL approaches, as explained in Section 17.2.2.

The first difference illustrates that in some respects, MDL is less restrictive than Bayes.[5] The second and third difference illustrate that in some respects, MDL is more restrictive than Bayes; yet, as I argue below, the MDL-imposed restrictions make eminent sense. In the remainder of this section, we first, in Section 17.2.1, explain the differences between MDL's luckiness approach and the Bayesian prior approach. Section 17.2.2 explains how MDL's insistence on data compression helps avoid some problems and interpretation difficulties with Bayesian inference. In Section 17.2.3 we discuss the relation between MDL and various sub-brands of Bayesian inference.

4. It is sometimes claimed that MDL inference has no subjective aspects. This is wrong: subjectivity enters MDL through the luckiness principle. This fact was somewhat hidden in earlier treatments of MDL, because "luckiness functions" were not made explicit there.

5. From a Bayesian point of view, one may dismiss this difference by claiming that LNML, two-part and plug-in codes should merely be viewed as approximations of the Bayesian universal code, which is what really should be used. The other two differences cannot be dismissed on such grounds.

17.2.1 Luckiness Functions vs. Prior Distributions

Let Θ represent some parametric model $\{P_\theta \mid \theta \in \Theta\}$. Suppose we are interested in parameter estimation relative to Θ, or in model selection between Θ and some other parametric model Θ°. To apply MDL inference, we impose a luckiness function $a(\theta)$. While this is related to adopting a prior $\pi(\theta)$, we now explain why it is really not the same. First of all, recall that the prior $\pi(\theta)$ corresponding to $a(\theta)$ is in general *not* proportional to $e^{-a(\theta)}$. Rather, it is given by the luckiness-tilted Jeffreys' prior $\pi(\theta) \propto \sqrt{\det I(\theta)} e^{-a(\theta)}$.

Three Reasons for Choosing a Particular Luckiness Function A second and more important difference with Bayesian priors is that we may choose a particular luckiness function for all kinds of reasons: first, it may indeed be the case that we choose a particular $a(\theta)$ because we have prior beliefs that data for which $a(\hat{\theta}(x^n))$ is large are improbable. A second reason for imposing a certain luckiness function is that it may make our universal codes mathematically simpler or more efficiently computable, so that our inference problem becomes tractable. A third reason to impose a particular luckiness function arises when we deal with an inference problem for which some region $\Theta' \subset \Theta$ is simply of no interest to us. We can then make the corresponding model selection problem a lot easier by imposing a luckiness function with large $a(\theta)$ for $\theta \in \Theta'$.

Example 17.3 Let Θ represent the Poisson model, given in the mean-value parameterization. Suppose, for the sake of argument, that data are distributed according to some $\mu^* \in \Theta$. Let $\Theta' = \{\mu \mid \mu > 1000\}$. It may be that μ^* represents the average time between two phone calls in a particular neighborhood. A phone company may be interested in estimating μ^* because it can optimize its resources if it has a better idea of the length of phone calls. If μ^* is very large, than P_{μ^*} also has large variance (the Poisson model satisfies $\mathrm{var}_{\mu^*}[X] = \mu^*$), and then knowing μ^* or a good approximation thereof may not be very useful, and will not save a lot of money. So, the company may be interested in good estimates of μ^*, but only if μ^* is small.

Only the first of these three reasons for using luckiness functions truly corresponds to the use of a prior in Bayesian statistics. The second reason – choosing a luckiness function for computational reasons – corresponds to the use of pragmatic priors, which, as explained further below, is standard Bayesian practice, yet, in my view, cannot be justified by Bayesian theory. From a Bayesian point of view, the third reason may seem strange, since

it mixes up probability and utility considerations.[6] From an MDL point of view, this is as it should be, as we now explain.

The Rationale of Luckiness Suppose we have chosen a luckiness function $a(\theta)$. We then observe data x^n, and end up with a luckiness ML estimator $\hat{\theta}_a(x^n)$ which achieves a certain luckiness value $a(\hat{\theta}_a(x^n))$. What are the consequences of this for decision-making? If $a(\hat{\theta}_a(x^n))$ is small, then our universal code $\bar{P}$ based on $a(\theta)$ achieved small regret. This means we are *lucky*: because we were able to compress the observed data a lot, we have a lot of confidence in any inferences we may draw from our universal code. For example, if we use the predictive distribution $\bar{P}(\cdot \mid x^n)$ as an estimator, then small $a(\hat{\theta}_a(x^n))$ means high confidence that $\bar{P}(\cdot \mid x^n)$ will lead to good predictions of future data. This conclusion can also be motivated from a frequentist perspective by Theorem 15.1, which relates good compression to fast learning. On the other hand, if the data were such that $a(\hat{\theta}_a(x^n))$ is large, then we were unlucky: we do not compress a lot, and we cannot trust predictions of future data based on our current predictive distribution $\bar{P}(\cdot \mid x^n)$ to be accurate. Another connection to frequentist analysis can be made in a model selection context, when comparing the model (Θ, P_θ) to a point hypothesis P_0. In that case, observing data with small $a(\hat{\theta}_a(x^n))$ implies that in a frequentist hypothesis test, we would have rejected P_0. This is implied by the no-hypercompression inequality, as we explained in Chapter 14, Example 14.6, page 421. It is crucial to note that *observing "lucky" data with small $a(\hat{\theta}_a(x^n))$ implies large confidence in estimates and predictions irrespective of whether our luckiness function corresponds to prior beliefs or not!* This shows that it is *safe* to choose a luckiness function completely at will – if it is chosen for mathematical convenience, and the corresponding prior puts large mass at parameter values that do not turn out to fit the data well, then we will simply conclude that we were "unlucky," and not have any confidence in our future predictions. We will see below that the same cannot be said for pragmatically chosen Bayesian priors.

Pragmatic Priors vs. Luckiness Functions: Expectation vs. Hope Most practical Bayesian inference is based on pragmatic priors, which, rather than truly representing the statistician's prior degree of belief, are chosen mostly

6. But see Rubin (1987), who mathematically shows that Bayesian prior and utility considerations are, in a sense, logically inseparable. This is, however, not the way Bayesian theory is usually presented.

for their mathematical convenience. An example is the frequent adoption in Bayesian practice of conjugate priors relative to exponential families (Berger 1985). One may argue that the use of such pragmatic priors corresponds exactly to the use of luckiness function in MDL priors. There is, however, a crucial difference. To see this, suppose we adopt some convenient, pragmatic prior π on the model Θ. For example, let Θ be the Bernoulli model in its standard parameterization, and suppose a pragmatic Bayesian adapts the uniform prior for convenience. Now the prior probability that θ will fall into the interval $[0, 0.9]$, is nine times as large as the prior probability that θ will fall in the region $[0.9, 1.0]$. Then strictly speaking, a Bayesian should be prepared to pay up to ten dollars for a lottery ticket which pays 100 dollars if, in a long sequence of outcomes, more than 90% ones are observed, and which pays 0 dollars otherwise. But why would this make sense if the uniform prior has been chosen for pragmatic, e.g. computational reasons, and does not *truly and precisely* reflect the Bayesian's subjective prior belief? To me, it does not make much sense. The same argument holds for most other decisions that can be made on the basis of pragmatic priors and posteriors.

Note that MDL inference with luckiness functions is immune to this problem. For example, suppose we use MDL with a Bayesian universal code relative to the Bernoulli model. We decide to use Jeffreys' prior (and use a uniform luckiness function), because it achieves codelengths that are close to minimax optimal. Now, once Jeffreys' prior has been adopted, one can formally calculate the prior probability that θ will fall into the interval $[0.5, 0.51]$, and the prior probability that θ will fall in the region $[0.99, 1.0]$. The latter probability is more than 5 times the former, but this certainly does not mean that an MDL statistician deems data with 99% or more 1s as a priori five times as likely than data with about 50% ones; he or she would certainly not be willing to take part in a betting game which would be fair if this proposition were true. From the MDL viewpoint, Jeffreys' prior has been adopted only because it leads to minimax optimal relative codelengths, *no matter what the data is*, and no statement about which parameters are "more likely" than others can ever be based on it.

According to MDL, all decision-making should be directly based on the universal code, which is a distribution on sequences of data. Thus, we can use the marginal likelihood $\bar{P}_{\text{Bayes}}(x^n)$ as a basis for decisions in model selection problems, and the predictive distribution $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$ as a predictor of future data or as an estimator of θ. We are only willing to place bets on events whose expected payoffs can be expressed directly in terms of

$\bar{P}_{\text{Bayes}}$. The luckiness-Jeffreys' posterior of θ has no meaning in and of itself.[7] Whereas in Bayesian inference, confidence in decisions is measured based on the posterior, in MDL it is solely measured by the amount of bits by which $\bar{P}_{\text{Bayes}}(x^n)$ compresses data x^n. Thus, with a very small sample, say, $n = 1$, $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$, is of course still strongly dependent on the luckiness function, and hence very "subjective," we are allowed to use it for prediction of X_2. But our confidence in the prediction is measured by the amount of bits by which $\bar{P}_{\text{Bayes}}(x^n)$ compresses data x^n relative to some null model $\mathcal{M}_0$ (Chapter 14, Example 14.3); for small n, this will usually give us very small confidence. For $n = 0$ (no data observed, distribution fully determined by luckiness function), it will give us no confidence at all — which is exactly how it should be, according to the luckiness principle.

Summarizing and rephrasing the previous paragraphs:

Bayesian Priors vs. MDL Luckiness Functions
From a Bayesian point of view, adopting a prior π implies that we a priori *expect* certain things to happen; and strictly speaking, we should be willing to accept bets which have positive expected pay-off given these expectations. For example, we always believe that, for large n, with high probability, the ML estimator $\hat{\theta}(x^n)$ will lie in a region Θ_0 with high prior probability mass. If this does not happen, a low-probability event will have occurred and we will be surprised.
From an MDL point of view, adopting a luckiness function a implies that we a priori *hope* certain things will happen. For example, we hope that the ML estimator $\hat{\theta}(x^n)$ will lie in a region with small luckiness function (i.e., high luckiness) $a(\theta)$, but we are not willing to place bets on this event. If it does not happen, then we do not compress the data well, and therefore, we do not have much confidence in the quality of our current predictive distribution when used for predicting future data; but we will not necessarily be surprised.

Bayes and Gzip In prequential MDL approaches, we predict X_{n+1} using $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$. Our predictions would be optimal in posterior expectation, *if*

7. Note that $\bar{P}_{\text{Bayes}}(x^n)$ and $\bar{P}_{\text{Bayes}}(x_{n+1} \mid x^n)$ can be written as expectations taken over the prior and the posterior, respectively. Thus, we cannot say that MDL considers *all* expectations over prior/posterior to be meaningless; the problematic ones are those which cannot be rewritten in terms of $\bar{P}_{\text{Bayes}}$ only.

data were really sampled by first sampling θ from the prior π_a, and then sampling data from θ. Does this imply that MDL secretly assumes that π_a is a prior in the Bayesian sense after all? Certainly not. To see why, consider the widely used data compression program *gzip*. Gzip is really just a prefix code, and by the Kraft inequality, there must be a (possibly defective) probability distribution $\bar{P}_{\text{gzip}}$ such that for all files, represented as a string x^n, the number of bits needed to encode x^n by gzip is given by $-\log \bar{P}_{\text{gzip}}(x^n)$. Thus, gzip would be the optimal code in expectation to use if data were actually sampled according to P_{gzip} or if, as a Bayesian would think of it, P_{gzip} would truly express our subjective uncertainty about the data. But it would be absurd to assume that either of these two is the case. This would imply that the entropy $H(P^{(n)}_{\text{gzip}})$ is a reliable estimate of the number of bits we would need for encoding an actual file x^n. Now some people use gzip only for the compression of pdf files, and others use gzip only for the compression of postscript files. These two groups achieve different compression rates, so for at least one of them, $H(P^{(n)}_{\text{gzip}})$ must be a very bad indicator of how well they can compress their files. Just like gzip can compress well even if one does not believe that $\bar{P}_{\text{gzip}}$ truly represents one's uncertainty, it is also the case that $\bar{P}_{\text{Bayes}}$ based on prior π_a can compress well, even if one does not believe that $\bar{P}_{\text{Bayes}}$, or equivalently, π_a truly represents one's uncertainty. $\bar{P}_{\text{Bayes}}$ may even be the best compressor one can think of, in the limited computation time that one has available. Note that we are really reiterating the "third-most important observation" of this book here, see Chapter 3, page 107.

This difference between MDL and Bayesian inference is also brought out if $\bar{P}_{\text{Bayes}}(x^n)$ cannot be computed, and we have to use an approximation instead. What constitutes a valid approximation? As we already described in Section 14.6 on page 453, an approximation which performs well apart for data with ML estimators that fall in a region with very small prior probability is acceptable from a Bayesian point of view, but not from an MDL (and not even from an expectation-based MDL) point of view.

Some Bayesians agree that, if a pragmatic convenience prior is used, then expectations defined with respect to the prior are not very meaningful, and even expectations over the posterior should be treated with some care. Yet the point is that, in contrast to MDL, there is nothing in Bayesian statistics which explicitly rules out taking such expectations. Moreover, many Bayesian procedures are explicitly based on taking such expectations. We give a particular example (DIC) below. Finally, even if a Bayesian admits that her prior may be "wrong," and only intends to use Bayes if the sample is so large that the prior hardly matters, there remains a huge conceptual problem if not just the prior, but the model itself is wrong. We very often want to use

such models which we *a priori* know to be wrong; see Example 1.6. If we use Bayes for such models, then we are forced to put a prior distribution on a set of distributions which we know to be wrong - that is, we have degree-of-belief 1 in something we know not to be the case. From an MDL viewpoint, these priors are interpreted as tools to achieve short codelengths rather than degrees-of-belief and there is nothing strange about the situation; but from a Bayesian viewpoint, it seems awkward (but see Section 17.2.3 on purely subjective Bayes).

DIC An example of a Bayesian procedure that explicitly relies on expectations over the posterior, even when the sample is small, is the Bayesian *deviance information criterion (DIC)* for model selection (Spiegelhalter, Best, Carlin, and van der Linde 2002). DIC is based on the posterior expected *deviance*

$$E_{\theta \sim w(\cdot|X^n)}[-\log P_\theta(X^n) + \log P_{\hat{\theta}_{\text{mean}}(X^n)}(X^n)], \tag{17.1}$$

where $w(\cdot \mid X^n)$ represents the posterior distribution of θ and $\hat{\theta}_{\text{mean}}$ is the posterior mean estimator. The latter is arrived at by taking expectations over the posterior (Chapter 15, Section 15.4.3). The examples in (Spiegelhalter et al. 2002) are based on standard, pragmatic priors (such as the normal prior for the linear model), which may be called "weakly informative" or "essentially flat" (see Bernardo's comment on (Spiegelhalter et al. 2002, page 625)). Such priors usually do not truly reflect the statistician's beliefs. Therefore, expectations taken over the posterior, and hence, the entire derivation leading up to the DIC criterion, are essentially meaningless from an MDL perspective.

Note that we chose this particular example because it is recent and directly related to model selection and regret; numerous other examples could be given as well.

17.2.2 MDL, Bayes, and Occam

Bayesian model selection can be done in various ways. One of the most straightforward and popular of these is the *Bayes factor* method (Kass and Raftery 1995). The Bayes factor method automatically implements a form of Occam's razor. This "Occam factor" phenomenon has been independently observed by several researchers; see (MacKay 2003, Chapter 28) for a detailed account, and (Jeffereys and Berger 1992) for a list of references. Below we shall see that in some other contexts such as nonparametric estimation and prediction, Bayesian inference can sometimes become disconnected from Occam's razor, and that this can cause trouble. Bayesian statisticians often view Bayesian inference as the more fundamental concept, and use the Occam

factor phenomenon to argue that a form of Occam's razor is implied, and therefore justified, by the deeper principle that rational learning and inference should be done in a Bayesian manner. In this subsection I argue that one may just as well turn the argument on its head and view Occam's razor, formalized as MDL inference based on universal coding, as the deeper principle. It justifies aspects of Bayesian inference by showing that they are implied by MDL's universal coding approach; but it also rules out some other, problematic aspects of Bayesian inference which contradict MDL ideas.

Bayes Factors: Direct Occam As we showed in Chapter 14, Section 14.2.3, page 418, Bayes factor model selection ends up being very similar to MDL model selection. This is all the more remarkable since the motivation for both methods is entirely different. Let us describe the motivation for the Bayes factor method. For simplicity, we restrict to the simple case where there are just two parametric models $\mathcal{M}_1$ and $\mathcal{M}_2$ under consideration. From a Bayesian perspective, we can assign each of these a prior probability W. Fortunately, the precise probabilities we assign will hardly affect the result, unless they are very close to zero or one, or the sample is exceedingly small. So, just for simplicity, we set $W(\mathcal{M}_1) = W(\mathcal{M}_2) = 0.5$. We now observe some data x^n and our task is to select a model $\mathcal{M}_1$ or $\mathcal{M}_2$. In the Bayes factor method, the goal of model selection is to find the true state of nature. This can be formalized by using a $0/1$-loss (minus utility) function $\mathbf{L} : \{1,2\}^2 \to \{0,1\}$: suppose $\mathcal{M}_\gamma$ is the "true" model, and we select model with index $\hat{\gamma}$. Then $\mathbf{L}(\gamma, \hat{\gamma}) := 0$ if $\gamma = \hat{\gamma}$ (our guess is correct), and $L(\gamma, \hat{\gamma}) := 1$ if $\gamma \neq \hat{\gamma}$ (our guess is wrong). According to Bayesian statistics, upon observing data $x^n \in \mathcal{X}^n$, we should take the decision $\hat{\gamma} \in \{1,2\}$ that minimizes the posterior expected loss, i.e. we set

$$\hat{\gamma} = \arg\min_{\gamma \in \{1,2\}} E_{Z \sim W(\cdot \mid x^n)}[\mathbf{L}(Z, \gamma)], \tag{17.2}$$

where

$$E_{Z \sim W(\cdot \mid x^n)}[\mathbf{L}(Z, \gamma)] = \sum_{\gamma' \in \{1,2\}} W(\mathcal{M}_{\gamma'} \mid x^n) \mathbf{L}(\gamma', \gamma).$$

Evidently, $\hat{\gamma}$ is achieved for the γ with the maximum posterior probability $W(\mathcal{M}_\gamma \mid x^n)$. This is exactly the γ picked by the Bayes factor method. In Section 14.2.3 we explained how to calculate $W(\gamma \mid x^n)$, and why it usually leads to the same results as MDL model selection.

The derivation makes clear why Bayes factors are sometimes criticized on the grounds that they strongly depend on one of the models being true in the

sense of being identical to the data generating process (Gelman et al. 2003), and that they have no "predictive interpretation"(Spiegelhalter, Best, Carlin, and van der Linde 2002).

Bayesian Model Selection and Prediction Our MDL analysis shows that both criticisms are unjustified: Bayes factor model selection can be viewed as a form of MDL model selection, which has a predictive (prequential) interpretation that is valid no matter what sequence is observed, and which does not depend on any underlying "true distribution."

Intriguingly however, Bernardo and Smith (1994, Chapter 6) consider alternative utility functions that correspond to viewing model selection in predictive rather than truth-hunting terms. They find that Bayesian model selection with such alternative utility functions behaves quite differently from Bayes factor model selection. More precisely, they replace the $0/1$ loss function in (17.2) by the log loss $-\log \bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n, \mathcal{M}_{\hat{\gamma}})$. Here $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n, \mathcal{M}_{\hat{\gamma}})$ is the Bayesian predictive distribution for the next outcome based on model $\mathcal{M}_\gamma$. Thus, they select the model $\hat{\gamma}$ such that the Bayesian prequential estimator based on $\mathcal{M}_{\hat{\gamma}}$ has the smallest posterior expected log loss. Their analysis suggests that asymptotically, the resulting "predictive Bayesian" model selection method will behave like leave-one-out cross-validation (Section 17.6) and the AIC criterion (Section 17.3.2). Also, unlike the Bayes factor method, their heuristic analysis still makes sense if the "true" data generating machinery is unknown, and none of the models under consideration is true. In this case, the models $\mathcal{M}_\gamma$ are simply viewed as sets of predictors. More precisely, they assume a "true" model class $\mathcal{M}^*$ and a prior W^* on $\mathcal{M}^*$, where $\mathcal{M}^*$ does not necessarily contain any of the $\mathcal{M}_\gamma$. Their analysis suggests that, under some conditions, asymptotically, someone who performs Bayesian model selection based on the loss function $-\log \bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n, \mathcal{M}_{\hat{\gamma}})$ relative to model class $\mathcal{M}^*$ and prior W^*, would select the same model $\mathcal{M}_{\hat{\gamma}}$ as would be selected by leave-one-out cross-validation, which can be implemented without knowledge of $\mathcal{M}^*$. Thus, while Bayes factor model selection has a *sequential* predictive interpretation, the Bayesian method suggested by Bernardo and Smith (1994) may have a leave-one-out style predictive interpretation.

Because of its correspondence to MDL model selection, it is clear that the Bayes factor method "automatically" implements some form of Occam's razor. If Bayes is used for estimation or prediction rather than model selection, then this connection may sometimes be lost. For example, in nonparametric estimation, the question whether or not Bayes has a built-in Occam's razor strongly depends on the chosen model and prior. Diaconis and Freedman (1986) provide combinations of models and priors for which Bayesian inference becomes inconsistent. As we argue below, this is really implied by the

fact that with such priors, the Bayesian universal code does not compress, and hence, does not implement a form of Occam's razor. On the other hand, the Gaussian processes that we introduced in Chapter 13 define models and priors for which the Bayesian universal code compresses very well, and, as a consequence, the Bayesian predictive distribution predicts exceedingly well. Taken together, these two facts illustrate why one might prefer the principles underlying MDL over those underlying Bayesian inference.

Gaussian Processes: Hidden Occam In nonparametric contexts, Bayesian prediction is sometimes based on a mixture of infinitely many arbitrarily complex distributions. Yet, in many such cases, Bayesian methods predict exceedingly well in practice. Therefore, it is sometimes argued that nonparametric Bayes violates the spirit of Occam's razor, but that also, this is as it should be. Take, for example, the Gaussian processes with RBF kernel. Regression based on such highly nonparametric models is very successful in practice (Rasmussen and Williams 2006). As we described in Section 13.5, such Gaussian process regression is based on a Bayesian predictive distribution which itself is essentially a mixture of infinitely many Gaussians. Therefore, it seems to violate Occam's razor.

I strongly disagree with this interpretation. When Gaussian processes are combined with the RBF kernel, then the Bayesian marginal likelihood has excellent, almost magical universal coding properties – the developments in Chapter 13, Section 13.5.3 show that it can be viewed as an excellent data compressor, with small coding regret even relative to high-dimensional regression functions. Using the prequential MDL convergence theorem, Theorem 15.1, this implies that, even if the data are distributed by some quite complex process, the Gaussian process predictions converge to the optimal predictions at a very fast (logarithmic) rate. Therefore, an MDL analysis of the Gaussian process model shows that, when used to sequentially code the data, it leads to very short descriptions thereof, and therefore does implement some form of Occam's razor after all; because good compression implies fast learning, it is exactly its good compression behavior which explains why it works so well in practice (a related point is made by (Rasmussen and Ghahramani 2000)).

In a nonparametric Bayesian modeling context, Neal (1996) states that:

> "For problems where we do not expect a simple solution, the proper Bayesian approach is therefore to use a model of a suitable type that is

as complex as we can afford computationally, regardless of the size of the training set."

Our analysis suggests that this is true *only* if the chosen model has good universal coding properties. We proceed to confirm this suggestion by reviewing an example with a model/prior combination that has bad universal coding properties, and for which Bayesian inference leads to bad results.

Bayesian Inconsistency: No Occam For some nonparametric i.i.d. model classes $\mathcal{M}$, there exist priors such that the corresponding Bayesian marginal likelihood $\bar{P}_{\text{Bayes}}$ is not a universal model relative to $\mathcal{M}$. More precisely, there exists $P^* \in \mathcal{M}$ and $c > 0$ such that, if data are i.i.d. $\sim P^*$, then no matter how large the sample n, the expected redundancy $E_{X \sim P^*(n)}[-\log \bar{P}_{\text{Bayes}}(X^n) + \log P^*(X^n)] > cn$. Thus, $\bar{P}_{\text{Bayes}}$ is not universal relative to P^*. Since MDL is *defined* as inference based on universal models, it is clear that from an MDL point of view, such priors can *never* be used.

Now interestingly, there exist some (in)famous theorems by Diaconis and Freedman (1986), which show that for some nonparametric contexts and with some priors, Bayesian inference can be inconsistent, in the sense that for some P^*, if data are i.i.d. $\sim P^*$, then the posterior concentrates on (smaller and smaller Hellinger neighborhoods of) a distribution P' with nonzero Hellinger distance to P^*. Bayesians often dismiss these results as irrelevant, since they are based on "silly" combinations of models and priors, that one would never use in practice. There is, however, nothing in standard Bayesian statistics which gives any clue about the conditions under which a model/prior combination is "silly." MDL provides exactly such a clue: it turns out that the combination of priors and true distributions used by Diaconis and Freedman are invariably such that the resulting $\bar{P}_{\text{Bayes}}$ is not a universal code relative to P^*. Thus, from an MDL point of view, one would never use such priors! To see this, note that Theorem 15.1 immediately implies the following: if $\bar{P}_{\text{Bayes}}$ is a universal code relative to P^*, then Bayes must be Césaro-consistent in KL, and therefore, also in Hellinger distance. The Diaconis-Freedman results imply that with the Diaconis-Freedman prior, the Bayesian predictive distribution $\bar{P}_{\text{Bayes}}$ is *not* Césaro consistent in Hellinger distance. It follows that under the Diaconis and Freedman prior, $\bar{P}_{\text{Bayes}}$ cannot be universal. In his comment on Diaconis and Freedman (1986), Barron (1986) already brings up a related point.

Bayesian vs. MDL Inconsistency
Suppose that data are sampled from a distribution P^* in the model class $\mathcal{M}$ under consideration. As a direct consequence of its focus on data compression, in prediction and estimation contexts, MDL is 100% immune to inconsistency by Theorem 15.1 and Theorem 15.3. In contrast, Bayesian inference can be inconsistent for some combinations of nonparametric $\mathcal{M}$ with some priors. In *model selection contexts*, both MDL and Bayes can be inconsistent in at least one case (Chapter 16, Section 16.4). If the model class is wrong, then MDL and Bayes can both be inconsistent in the sense of (Grünwald and Langford 2004).

17.2.3 MDL and Brands of Bayesian Statistics

In the previous subsections, we criticized the "pragmatic" version of Bayesian statistics that is most often used in practice. But perhaps Bayes was never intended to be used in such a pragmatic way; if we use Bayesian inference as it was intended by its founding fathers, then maybe our criticisms become invalid. Matters are complicated by the fact that there exist various brands of pure, "nonpragmatic" Bayesian statistics. Here we examine the main brands, (purely) subjective Bayes, mostly associated with the work of De Finetti (1937) and Savage (1954), and objective Bayes, mostly associated with Jeffreys (1961) and Jaynes (2003). We also look into Solomonoff's (1964) approach, a specific version of objective Bayes that forms a middle ground between Bayes and MDL. Below, we characterize each brand by the type of prior that it uses.

Nonpragmatic Subjective Priors If a decision-maker thinks about a problem long enough, then she may avoid pragmatic priors and instead use priors that truly reflect her degrees of belief that various events occur. Assuming that the decision-maker can come up with such priors, some of the problems with Bayes that we mentioned above will disappear. In particular, expectations taken over the prior will be meaningful, at least in a subjective sense. Based on the "Dutch Book Argument" of De Finetti (1937), or on the axiomatic approach of Savage (1954), one may claim that subjective Bayes is the only rational ("coherent") approach to decision-making and should be preferred over MDL; in this view, the problems we mentioned are all due to the use of pragmatic priors. I have two problems with this position. First,

while I find De Finetti's and Savage's results very interesting, I am not at all convinced that they imply that a rational decision-maker should act like a Bayesian.[8] Second, I do not think that humans have sufficient imaginative power to come up with a prior distribution that truly represents their beliefs, as the following example illustrates.

Example 17.4 [Mercury's Perihelion Advance] Suppose that, after having observed data x^n, you selected some model $\mathcal{M}_0$ as a best explanation of the data. Later you learn that another research group found an entirely different explanation $\mathcal{M}_1$ of the data, such that

$$-\log \bar{P}_{\text{Bayes}}(x^n \mid \mathcal{M}_1) \ll -\log \bar{P}_{\text{Bayes}}(x^n \mid \mathcal{M}_0).$$

If you can be reasonably sure that $\mathcal{M}_1$ has not been constructed with hindsight, after seeing data x^n (so that there was no "cheating" going on), then you may well want to abandon the model $\mathcal{M}_0$ in favor of $\mathcal{M}_1$, *even if you had put no prior probability on* $\mathcal{M}_1$ *in the first place*. According to the subjective Bayesian viewpoint, this is not possible: if your prior probability of $\mathcal{M}_1$ was 0, the posterior is 0, and you can never embrace $\mathcal{M}_1$. But how can you put a prior on all possible models $\mathcal{M}_1$? Surely the imagination of individuals and research groups is limited, and they cannot think of all possible explanations – which may be provided by other research groups – in advance. For example, it was discovered in the 19th century that Mercury's perihelion does not exactly follow the predictions of Newton's theory of gravitation. As astronomers gathered more and more data about this phenomenon, various explanations ("models") were suggested, such as the existence of an unknown planet "Vulcan." The matter was finally settled when it turned out that Einstein's theory of general relativity, discovered only in 1916, explained Mercury's perihelion perfectly well. If astronomers had been using subjective priors on models, then it seems quite unlikely that, before 1916, anyone except Einstein would have had a nonzero prior probability on the theory of general relativity. This implies that they would not have believed this theory after 1916, no matter how well it had accounted for the data. This problem is closely related to the "old evidence" problem in Bayesian confirmation theory (Hutter 2006).

In the MDL approach, we can effectively avoid this "zero prior problem." We mentioned this possibility already in Chapter 14, Section 14.4.4, page 436; see also Example 17.5. For example, suppose we want to select a model $\mathcal{M}_\gamma$ from a CUP model class $\mathcal{M} = \cup_\gamma \mathcal{M}_\gamma$, $\gamma \in \Gamma$. We can use a meta-two-part code in which $\dot{L}(\gamma)$, the codelength function for γ, corresponds to a defective distribution W that sums to $1/2$ rather than 1. If later somebody proposes a new model

8. To mention just one, of many, problems: in my opinion, the *Ellsberg* paradox convincingly shows that sometimes uncertainty about an event cannot be represented by a single number; see (Ellsberg 1961) and (Halpern 2003, Example 2.3.2).

$\mathcal{M}_a, a \notin \Gamma$ for the data x^n that we were trying to model, and we are confident that that person has not peeked at x^n, then we can assign a code word with length $\dot{L}(a) = 1/4$ to $\mathcal{M}_a$; now $\dot{L}$ corresponds to a distribution W summing to $3/4$ rather than $1/2$. Now if another trustworthy person comes along and proposes some model $\mathcal{M}_b$, $b \notin \Gamma$, we can set $\dot{L}(b) = 1/8$; now W sums to $7/8$. The process can be repeated infinitely often. From an MDL perspective, there is nothing strange about this procedure. From a Bayesian point of view, however, it seems awkward: if we use a defective prior such as W before we are told about any of the alternative models $\mathcal{M}_a$, $\mathcal{M}_b, \ldots$, then we are effectively basing our decisions on the posterior $W(\mathcal{M}_\gamma \mid x^n, \gamma \notin \{a, b, \ldots\})$. Thus, it seems as if we have already decided that the true state of the world is a model $\mathcal{M}_\gamma$ on the original list, with $\gamma \in \Gamma$. If we are then told to consider the newly proposed model $\mathcal{M}_a$, we would have to "decondition" and assume that $\mathcal{M}_a$ may contain the true state of the world after all.

Solomonoff's Nonpragmatic, "Universal" Priors In the earliest version of what we called "idealized MDL" in Chapter 1, Section 1.4, Solomonoff (1964) proposed a prior $\mathbf{M}$ on a countable set $\mathcal{M}$ that includes all computable probabilistic sources. Broadly speaking, this prior assigns large mass to sources P that can be implemented by a short computer program. More precisely, there is some constant C such that for all computable sources P, $-\log \mathbf{M}(P) \leq K(P) + C$, where $K(P)$ is the length of the shortest computer program that, when input n, x^n and precision d, outputs the first d bits of $P(x^n)$ and then halts. It is natural to extend the definition of Kolmogorov complexity and call $K(P)$ the Kolmogorov complexity of the distribution P (Li and Vitányi 1997). Here we consider Solomonoff's approach from a Bayesian perspective. We will have more to say about its relation to (practical, nonidealized) MDL in Section 17.8.

One can show that the Bayesian universal model $\bar{P}_{\text{Solomonoff}}$ relative to such a prior is $O(1)$-universal relative to any other computable probabilistic source, including any other universal model (Li and Vitányi 1997). Since we may assume that in practice, any universal model we will ever use in MDL or Bayesian inference is computable, $\bar{P}_{\text{Solomonoff}}$ may serve as a replacement for any other universal model that we might be interested in. Similarly, it may be reasonable for a decision-maker to use the prior $\mathbf{M}$ as a proxy of the subjective prior that he would really like to use, but cannot formulate for lack of time and imagination. If $\mathbf{M}$ is used in this way, then the problem of zero prior (Example 17.4) disappears: $\mathbf{M}$ assigns positive prior mass on any computable theory that can be formulated at all. Hutter (2006) forcefully

argues that **M** solves various other problems of subjective Bayesianism as well.

Still, this approach is not without its problems. As mentioned in Chapter 1, Solomonoff's and other idealized MDL approaches are inherently uncomputable, and it is not clear whether sufficiently general computable approximations exist. Second, if we apply it to small samples (as, in statistical practice, we often must),[9] then the choice of programming language on which it is based has a significant impact on the Bayesian predictive distribution. We may choose the language that best suits the phenomenon that we are trying to model, but then subjectivity enters again, and the problems of the strictly subjective approach reappear.

Finally, it seems that using Solomonoff's approach, we violate the weak prequential principle. For example, it is not clear how we can use it for universal prediction in the weather forecasting example (Example 17.2), where the Kolmogorov complexity of the various forecasters is unknowable.

The relation between MDL inference, subjective priors, and Solomonoff's objective priors, is discussed further in Example 17.5.

Other "Objective" Priors In the *objective Bayesian* approach, one replaces subjective or pragmatic priors by more "objective" ones, which may be used if no or very little prior knowledge is available. Examples are the Jeffreys' prior (Jeffreys 1961), and Bernardo's *reference priors*, which sometimes, but not always, coincide with Jeffreys' prior (Bernardo and Smith 1994). Various other possibilities are explored by Berger and Pericchi (2001). In contrast to Solomonoff's approach, the focus here is usually on parametric estimation or model selection between parametric models, and the priors are computable in practice. It is generally recognized (even by Jeffreys and Bernardo) that the choice of "objective" prior should depend on, for example, the loss function of interest: a prior which works as a good proxy for an unknown prior in one context (e.g., model selection), may not be such a good proxy in another context (e.g., parameter estimation). Some objective Bayes approaches to Bayes factor model selection are almost identical to MDL approaches with Bayesian universal codes.

In my view, there is still some advantage in interpreting these procedures from an MDL universal coding point of view, rather than as Bayesian. When

9. Consider, for example, experiments where each subject has to be paid in order to take part. In such experiments, one often has to make do with about 30 sample points. These are common in, e.g., the field of psychology.

Jim Berger recently (2005) gave an excellent tutorial on objective Bayes methods in Amsterdam, a member of the audience asked: "In both frequentist methods and subjective Bayesian methods, the interpretation of probability is clear, albeit very different. But how should we interpret the probabilities appearing in objective Bayesian methods?" Berger answered: "In many cases, both interpretations are valid." My answer would have been: "In *all* cases, these probabilities can be interpreted as *codelengths!*"

17.2.4 Conclusion: a Common Future after All?

In the previous section, I have criticized several aspects of the Bayesian approach. I do, however, see problems with the MDL approach as well — such as the lack of a proper accompanying decision theory; see Section 17.11 — and I do recognize that the two approaches are often similar in practice. As said, MDL methods often resemble objective Bayes methods. I should also mention that there is a nontrivial overlap between Rissanen's individual-sequence philosophy and De Finetti's original motivation for subjective Bayesian analysis, which is still shared by some Bayesians. Such subjectivists prefer not to speak about nonobservable things such as "θ, the true probability, or long-term frequency of heads." (Diaconis 2003). In fact, the preface of the magnum opus (De Finetti 1974) opens with the following memorable lines:

> "My thesis, paradoxically, and a little provocatively, but nonetheless genuinely, is simply this: PROBABILITY DOES NOT EXIST.
>
> The abandonment of superstitious beliefs about the existence of Phlogiston, the Cosmic Ether, Absolute Space and Time, . . ., or Fairies and Witches, was an essential step along the road to scientific thinking. Probability, too, if regarded as something endowed with some kind of objective existence, is no less a misleading misconception [. . .]."

Subjectivists such as De Finetti only want to speak about probabilities of events that will eventually be observed (such as the probability that it will rain tomorrow), and they interpret these as degrees of belief, operationalized as the amount of money one is willing to place on certain bets. This is quite similar to Rissanen's ideas, who also restricts probabilist statements to observable events, and regards them as indicating the code one would use to code the data if the goal were to compress data as much as possible.

> In practice, subjective Bayesians then often speak about quantities like "θ, the true bias of the coin" after all, motivated either by De Finetti's exchangeabil-

ity theorem, or by Savage's treatment of subjective probability, which show that under certain circumstances, a rational decision maker should act *as if* data were distributed accorded to some $P_\theta \in \mathcal{M}$. From this point of view, individual-sequence MDL is like De Finetti-type subjectivist Bayesian statistics, but even more radical: any talk of a true θ is now truly avoided, and no inference is based on the presumption that such a θ exists.

Given these considerations, it may certainly be possible that one day, Bayesian and MDL approaches to statistics will merge into one. In such a joint approach, the contribution of MDL theory could be a proper interpretation of probabilities and expectations when pragmatic priors (now viewed as luckiness functions) are used, as well as some restrictions on the priors in nonparametric problems.

17.3 MDL, AIC and BIC

In this section, we specialize to MDL model selection. We compare it to two popular benchmark model selection methods: the *Bayesian Information Criterion* (BIC) and the *Akaike Information Criterion*, both of which we already encountered in Chapter 14, Example 14.4, page 417, and, in a regression context, in Section 14.5.4 on page 450.

17.3.1 BIC

In the first paper on MDL, Rissanen (1978) used a two-part code and showed that, asymptotically, and under regularity conditions, the two-part code-length of x^n based on a parametric model $\mathcal{M}_\gamma = \{P_\theta \mid \theta \in \Theta_\gamma\}$ with k_γ parameters, using an optimally discretized parameter space is given by

$$-\log P_{\hat{\theta}_\gamma(x^n)}(x^n) + \frac{k_\gamma}{2}\log n, \tag{17.3}$$

where $\hat{\theta}_\gamma$ is the ML estimator within Θ_γ, and $O(1)$-terms (depending on k_γ, but not on n) are ignored. As we have discussed in Chapter 14, these terms can be quite important in practice. In the same year Schwarz (1978), ignoring $O(1)$-terms as well, showed that, for large enough n, Bayes factor model selection between two exponential families amounts to selecting the model minimizing (17.3). As a result of Schwarz's paper, model selection based on (17.3) became known as the *BIC (Bayesian Information Criterion)*. Not taking into account the functional form of the model $\mathcal{M}$, it often does not work very well in practical settings with small or moderate samples.

Consistency BIC does tend to perform well if the sample size gets really large: Suppose BIC is used to select a model $\mathcal{M}_\gamma$ coming from some CUP model class $\mathcal{M} = \cup_{\gamma\in\Gamma}\mathcal{M}_\gamma$ (Case 1(c) of Chapter 16, Section 16.1.1). Then BIC is asymptotically consistent under a wide variety of conditions on the model class $\mathcal{M}$. Here we refer to consistency in the sense of Chapter 16, Section 16.3.1. Often, consistency will hold for *all* P^* in *all* $\mathcal{M}_\gamma$: there is no need to exclude subsets of measure 0, as there was with MDL model selection based on $\text{CUP}(\text{2-p}, \text{Bayes})$-codes (Barron, Rissanen, and Yu 1998; Csiszár and Shields 2000).

On the other hand, consider the nonparametric context where data are distributed according to some $P^* \in \mathcal{M}^*$, $P^* \notin \mathcal{M}$, where $\mathcal{M}^*$ is some smooth subset of $\langle\mathcal{M}\rangle$ (Case 2(a) of Chapter 16, Section 16.1.1). Then BIC model selection combined with maximum likelihood inference in the chosen model often does not achieve the minimax rate of convergence (Speed and Yu 1993; Yang 2005a; Yang 2005b). More precisely, the model selection-based estimator based on BIC and ML (defined analogously to (16.2) on page 509) typically does converge in terms of Hellinger risk for all $P^* \in \mathcal{M}^*$, but the rate of convergence exceeds the minimax rate by a factor of $\text{ORDER}(\log n)$; see (Yang 2005b, Section 4) and (Shao 1997). Here the "minimax rate" is the minimax optimal rate where the minimum is over all estimators, i.e. all functions from $\mathcal{X}^*$ to $\mathcal{M}$, and the maximum is over all $P^* \in \mathcal{M}$. This minimax rate is often of the form $n^{-2s/(2s+1)}$, where s is the degree of smoothness of the distributions in P^*; see Chapter 16, Example 16.4.

17.3.2 AIC

The *Akaike Information Criterion* was introduced and developed by Akaike in a series of papers starting with (Akaike 1973). It can be used for model selection with finite or countably infinite CUP model classes $\mathcal{M} = \cup_{\gamma\in\Gamma}\mathcal{M}_\gamma$. For given data x^n, it tells us to select the model with index γ_{aic} that minimizes, over all $\gamma \in \Gamma$,

$$-\log P_{\hat{\theta}_\gamma(x^n)}(x^n) + k_\gamma. \tag{17.4}$$

If $\mathcal{M}$ is countably infinite, then for large n, this criterion tends to select more complex models than BIC, the reason being that the complexity penalty does not depend on n.

Consistency The consistency properties of AIC are quite different from those of BIC. In the case where data are distributed according to some $P^* \in$

$\mathcal{M}_{\gamma^*}$ for some $\gamma^* \in \Gamma$ (Case 1(c) of Chapter 16, Section 16.1.1), AIC is often inconsistent (Shibata 1976; Hannan 1980; Woodroofe 1982). Recall from the previous subsection that BIC is typically consistent in such cases. On the other hand, consider the nonparametric context where data are distributed according to some $P^* \in \mathcal{M}^*$, $P^* \notin \mathcal{M}$, where $\mathcal{M}^*$ is some smooth subset of $\langle \mathcal{M} \rangle$ (Case 2(a) of Chapter 16, Section 16.1.1). Then, under a variety of conditions on $\mathcal{M}^*$, AIC combined with maximum likelihood estimation is not only consistent; it also converges at the minimax rate of convergence (Speed and Yu 1993; Yang 2005a; Yang 2005b). Relatedly, when used for predictions, the AIC-MDL squared prediction error converges to 0 at the minimax optimal rate (Shao 1997; Li 1987). Recall that in such cases, BIC is often slower than minimax by a factor of $\text{ORDER}(\log n)$. The upshot is that there exist cases for which BIC is asymptotically optimal whereas AIC is not, and vice versa.

The reason why AIC achieves these optimal convergence rates can be seen from a reinterpretation of Akaike's original derivation of AIC. In this reinterpretation, AIC is really an easily computable approximation of another criterion which we will call AIC*. AIC* is defined as the model selection criterion γ (see page 509) that, for each n, achieves

$$\max_{P^* \in \mathcal{M}^*} \min_{\gamma: \mathcal{X}^n \to \Gamma} E_{X^n \sim P^*}[D(P^* \| P_{\hat{\theta}_{\gamma(X^n)}})]$$

where the minimum is over all model selection criteria, i.e. all functions from $\mathcal{X}^*$ to Γ, and $\mathcal{M}^* \subset \langle \mathcal{M} \rangle$ is a subset of $\mathcal{M}$'s information closure that satisfies some smoothness conditions. By definition, AIC* attains the minimax optimal KL risk among all model selection-based estimators that use the ML estimator within the selected model. This makes it plausible (but by no means proves) that, under some conditions on $\mathcal{M}$ and $\mathcal{M}^*$, AIC, which can be seen to be an approximation of AIC*, achieves the minimax convergence rate among *all* (and not just model-selection based) estimators.

AIC achieves the minimax optimal convergence rate under certain regularity conditions on $\mathcal{M} = \cup_{\gamma \in \Gamma} \mathcal{M}_\gamma$ and $\mathcal{M}^*$. For example, these hold if, for each γ, $\mathcal{M}_\gamma$ is a linear model (Chapter 12) with γ covariates, (X_i, Y_i) are i.i.d., and $\mathcal{M}^*$ satisfies certain smoothness assumptions. They also hold for a wide variety of other models, including some time series models. However, we do have to restrict $\mathcal{M}^*$, i.e. we have to make assumptions about the $P^* \in \langle \mathcal{M} \rangle$ that is supposed to have generated the data; see (Yang and Barron 1999). Various modifications of AIC have been proposed for the case where such conditions on $\mathcal{M}$ and $\mathcal{M}^*$ are violated, or for the case of small samples. For the latter, see (Burnham and Anderson 2002).

We further note that if all the appropriate conditions hold, then the AIC-based estimator achieves the minimax optimal convergence rates both if (a) data are distributed according to some $P^* \in \mathcal{M}_{\gamma^*}$ for finite $\gamma^* \in \Gamma$ (this holds even if Γ is itself finite); and also if (b) $P^* \in \mathcal{M}^*, P^* \notin \mathcal{M}$. In case (a), the rate is $O(1/n)$. Also, under the same conditions, even though it achieves the minimax optimal convergence rate, AIC is inconsistent if $P^* \in \mathcal{M}_{\gamma^*}$ for some finite γ^*, both if Γ is finite and if Γ is countably infinite. In both cases, the inconsistency is of a curious type: as n increases, the P^*-probability that $\gamma_{\text{aic}}(X^n) > \gamma^*$ goes to some number p with $0 < p < 1$. Thus, AIC is only inconsistent with a certain probability; this probability neither tends to 1 nor to 0.

17.3.3 A Version of MDL that Combines the Best of AIC and BIC

AIC As Expectation-Based MDL From the *expectation-based* MDL point of view, the AIC idea makes a lot of sense. Indeed, it defines a prequential coding system, given by

$$\bar{P}_{\text{mdl-aic}}(X_{n+1} \mid x^n) := P(X_{n+1} \mid \hat{\theta}_{\gamma_{\text{aic}}(x^n)}(x^n)). \tag{17.5}$$

Under the conditions for which AIC is a good approximation of AIC*, $\bar{P}_{\text{mdl-aic}}$ will actually be a very good universal code relative to $\mathcal{M}^*$. Namely, by its definition, AIC* chooses the model index γ such that the worst-case expected codelength of encoding a new value X_{n+1} using the code with lengths $-\log P_{\hat{\theta}_\gamma(x^n)}(X)$ is minimized. If we were to look for the prequential plugin code relative to $\mathcal{M}$, that, assuming $P^* \in \mathcal{M}^*$, asymptotically achieves the minimax optimal redundancy, and under the constraint that the estimator to be used in the plugin code is model-selection based, then we would arrive at almost the same estimator/universal code as (17.5), but with two differences: (1) γ_{aic} is replaced by γ_{aic^*}; and, (2), rather than predicting by the ML estimator $\hat{\theta}_\gamma$ for the selected γ, it would be better to predict using a prequential MDL estimator $\bar{P}(\cdot \mid \mathcal{M}_\gamma)$ defined relative to $\mathcal{M}_\gamma$, for example, the Bayesian universal code with the luckiness-tilted Jeffreys' prior. Presumably however, such a modification would not affect the minimax convergence rate of $\bar{P}_{\text{mdl-aic}}$.

MDL Is Not BIC... It has sometimes been claimed that MDL = BIC; for example, Burnham and Anderson (2002, page 286) write "Rissanen's result is equivalent to BIC." This is wrong, even for the 1989 version of MDL that Burnham and Anderson refer to – as pointed out by Foster and Stine (2005), the BIC approximation only holds if the number of parameters k is kept

fixed and n goes to infinity. If we select between nested families of models where the maximum number of parameters k considered is either infinite or grows with n, then model selection based on both $\text{CUP}(\text{2-p}, \text{Bayes})$ and $\text{CUP}(\text{2-p}, \text{nml})$ tends to select quite different models than BIC; if k gets closer to n, the contribution to $\text{COMP}^{(n)}(\mathcal{M})$ of each additional parameter becomes much smaller than $(1/2)\log n$ (Foster and Stine 2005). Similarly, in $\text{CUP}(\text{2-p}, \text{2-p})$-MDL, if the discretized value of a parameter chosen by two-part MDL is close to 0, then, at least for some models, again the MDL procedure may actually behave more like AIC than BIC; see Foster and Stine (1999). Similarly, Hansen and Yu (2001) show that some versions of MDL model selection in linear regression punish even less for complexity than AIC (Chapter 14, page 450) for some data and models, and Hansen and Yu (2002) conjecture that their gMDL procedure actually combines the strengths of AIC and BIC. They prove gMDL consistent for the situation that $P^* \in \mathcal{M}_{\gamma^*}$ (Case 1(c) of Chapter 16), thus showing it has the strength of BIC; but the statement about AIC is only based on experimental results.

> We note that researchers who claim MDL = BIC do have an excuse: in early work, Rissanen himself has used the phrase "MDL criterion" to refer to (17.3), and, unfortunately, the phrase has stuck.

Sometimes, MDL Is Asymptotically BIC After All We just indicated that in many situations, at least some variations of $\text{CUP}(\text{2-p}, \cdot)$-code MDL model selection seem to behave more like AIC than BIC. But unfortunately, the fact remains that in other situations, asymptotically $\text{CUP}(\text{2-p}, \text{Bayes})$-code MDL model selection does not achieve the minimax optimal rate of convergence, whereas AIC does. As we remarked in Chapter 16, in some nonparametric contexts such as histogram density estimation (Case 2(a) of Chapter 16), prequential MDL estimation based on the $\text{CUP}(\text{Bayes}, \text{Bayes})$-codes, as well as MDL model selection based on $\text{CUP}(\text{2-p}, \text{Bayes})$-codes, does not achieve the minimax rate of convergence. Like BIC, it is too slow by a factor of $\text{ORDER}(\log n)$.

> **Open Problem No. 17** Although this log-factor is probably not that relevant for practical applications, we do consider this a serious issue: the fact that universal codes designed in standard ways such as $\text{CUP}(\text{2-p}, \text{Bayes})$ are sometimes not asymptotically optimal, is not so surprising. What is more worrying is that nobody knows how to design an alternative *individual-sequence* universal code based on $\mathcal{M}^*$, that, when used for model selection and analyzed in expectation under some $P^* \in \mathcal{M}$, does achieve the minimax optimal rate. It seems

that the common strategies for designing universal codes – NML, two-part, Bayes, plugin – all fail here, and something entirely new is needed.

Although there is no known individual-sequence based universal code that achieves the minimax optimal rate in all cases where AIC is known to achieve this rate, there does exist an expectation-based universal code that achieves it. This is just the code $\bar{P}_{\text{mdl-aic}}$ that we described above. Using this code for prequential estimation will lead to an estimator that, like AIC, is sometimes inconsistent if $P^* \in \mathcal{M}$. On the other hand, $\text{CUP}(\text{2-p}, \text{Bayes})$-estimators, which may not achieve the minimax optimal rate, are consistent if $P^* \in \mathcal{M}$ (Section 16.3). We now sketch a variation of MDL that is likely to combine the best of both worlds – although we have not formally proven that it does. Let, for the given model class $\mathcal{M}$, $\bar{P}_{\text{mdl-aic}}$ be the universal code that embodies AIC, as described above. Let $\bar{P}_{\text{CUP(2-p,Bayes)}}$ be a $\text{CUP}(\text{2-p}, \text{Bayes})$-universal code, where $\bar{P}_{\text{Bayes}}(\cdot \mid \gamma)$ is a Bayesian universal code relative to some given luckiness function or prior. We define a new, nonprequential code as follows. For any data sequence x^n, we first encode whether $-\log \bar{P}_{\text{mdl-aic}}(x^n)$ or $-\log \bar{P}_{\text{CUP(2-p,Bayes)}}$ is smaller. This takes one bit. We then encode x^n using whichever of the two codes compresses x^n more. The first bit can be thought of as part of our hypothesis: if $-\log \bar{P}_{\text{mdl-aic}}(x^n) < -\log \bar{P}_{\text{CUP(2-p,Bayes)}}$, this can interpreted as stating the hypothesis that "we are in a nonparametric situation; AIC is better." If the reverse inequality holds, this can be interpreted as stating the hypothesis "parametric situation; $\text{CUP}(\text{2-p}, \cdot)$-MDL is better." In the first case, model selection should proceed by AIC; in the second case, it should proceed by our $\text{CUP}(\text{2-p}, \cdot)$-MDL code. In essence, we are doing MDL model selection between two-universal codes, just as in Chapter 14. But the two universal codes now represent the hypotheses $P^* \in \mathcal{M}^* \setminus \mathcal{M}$ vs. $P^* \in \mathcal{M}$.

We do not regard this new universal code as a perfect solution to the AIC-BIC dilemma, since it only has an expectation-based, and not an individual-sequence MDL interpretation. It does illustrate, however, the power of basing learning on universal data compression – we simply use the fact that any two universal codes (in this case, the MDL-AIC code, which is optimal in nonparametric settings, and the $\text{CUP}(\text{2-p}, \text{Bayes})$-code, which is optimal in parametric settings) can be trivially combined into a new universal code that, on any given sequence performs essentially as well as the code that is best on that sequence.

It is sometimes claimed that the question whether to prefer AIC or BIC is irrelevant, since the two procedures have been developed with different goals

in mind – optimal prediction of future data vs. hunting for the true model, containing P^*; see, for example, (Forster 2001, page 90) and (Sober 2004, page 649). From this perspective, it may seem to be impossible or irrelevant to craft new methods that combine the strengths of AIC and BIC. Yet, BIC is an approximation of Bayes factor model selection, and as we showed in Section 17.2.2, Bayes factor model selection has a very clear predictive interpretation as well – it can be thought of as a prequential MDL method. Thus, both AIC and BIC are approximations to procedures with predictive interpretations and this suggests that it may be both possible and desirable to combine the strengths of both procedures after all. We are not the first to notice this: De Luna and Skouras (2003) propose a somewhat similar *model meta-selection method* which holds promise to asymptotically combine the best of both methods. Such a method was used earlier by Clarke (1997), and was extended by Clarke (2003), who also provides a theoretical analysis. Yang (2005b) uses a form of cross-validation to select between AIC and BIC and proves that, in a certain sense, it achieves the best of both worlds. This is a subtle issue though – Yang (2005a) shows that it is only possible to "combine the strengths of AIC and BIC" under a restricted definition of what exactly one means by "combining the strengths."

17.4 MDL and MML

MDL shares some ideas with the *Minimum Message Length (MML) Principle* which predates MDL by 10 years. Some key references are (Wallace and Boulton 1968; Wallace and Boulton 1975; Wallace and Freeman 1987) and (Wallace 2005); a long list of references is in (Comley and Dowe 2005). Just as in MDL, MML chooses the hypothesis minimizing the code-length of the data. But the *codes* that are used are quite different from those in MDL. First of all, in MML one always uses two-part codes, so that, like two-part code MDL, MML automatically selects both a model family and parameter values. Second, while MDL codes such as $\bar{P}_{\mathrm{nml}}$ minimize *worst-case (minimax)*, expected or individual sequence, *relative* code-length (i.e. redundancy or regret), the two-part codes used by MML are designed to minimize *a priori* expected *absolute* code-length. Here the expectation is taken over a subjective prior distribution on the collection of models and parameters under consideration; see the summary in Figure 17.1 on page 562. Note that such an approach flagrantly contradicts Rissanen's individual-sequence MDL philosophy: first, it is based on expectation rather than individual sequences; second, it is based on expectations taken over a prior distribution, which, as we explained in Section 17.2.1, cannot be justified from an MDL perspective — nevertheless, in practice it often leads to similar results.

Indeed, Wallace and his co-workers stress that their approach is fully (subjective) *Bayesian*. Strictly speaking, a Bayesian should report his findings by citing the full posterior distribution. But, as we explained in Chapter 15, Section 15.4.2, sometimes one is interested in a single model, or hypothesis for the data. In that case, Bayesians often use the MAP (Maximum A Posteriori) hypothesis; or the posterior mean parameter vector; or the posterior median. The first two approaches were described in Chapter 15, Section 15.4.3. As explained by Wallace (2005), all three approaches have some unpleasant properties. For example, the MAP and the mean approach are parameterization dependent. The posterior mean and median approaches cannot be used if different model families are to be compared with each other. The MML method provides a method for Bayesian estimation that avoids most of the problems of these standard methods.

Below we describe the main ideas behind MML in more detail, and we compare them to corresponding notions in MDL.

17.4.1 Strict Minimum Message Length

Let $\mathcal{M}$ be some given model class of probabilistic sources. MML takes a subjective Bayesian approach, and assumes that the statistician is able to formulate a subjective prior distribution W on the given model class $\mathcal{M}$, representing his subjective beliefs about the domain under consideration. $\mathcal{M}$ is usually either a parametric model or a CUP model class $\cup_{\gamma\in\Gamma}\mathcal{M}_\gamma$. In the latter case, W will be a hierarchical prior, consisting of a discrete distribution on Γ, and, for each $\gamma \in \Gamma$, a prior on the parametric model $\mathcal{M}_\gamma = \{P_\theta \mid \theta \in \Theta_\gamma\}$, given by some density $w(\theta \mid \gamma)$.

The basic idea behind MML modeling is then to find (a) a two-part description method and (b) an associated estimator, minimizing the *expected* two-part description length of the data. Here the expectation is taken according to the statistician's subjective distribution of X^n, which is just the Bayesian marginal likelihood $\bar{P}_{\text{Bayes}}$, defined with respect to the prior W. Formally, let $\dot{\mathcal{L}}$ be the set of partial codes for $\mathcal{M}$. If for some $\dot{L} \in \dot{\mathcal{L}}$, some $P \in \mathcal{M}$ cannot be encoded under $\dot{L}$, then we write $\dot{L}(P) = \infty$. For each n, for each code(length function) $\dot{L}$ in the set $\dot{\mathcal{L}}$, we can examine the expected length of the corresponding two-part code, where the two-part code is defined just as in Chapter 10, (10.1), page 272 (if $\mathcal{M}$ is parametric), and more generally, in

Chapter 15, (15.14), page 477:

$$E_{P\sim W}E_{X^n\sim P}[\min_{\ddot{P}\in\mathcal{M}}\{\ \dot{L}(\ddot{P})-\log\ddot{P}(X^n)\ \}\ ]=$$
$$E_{X^n\sim\bar{P}_{\text{Bayes}}}[\min_{\ddot{P}\in\mathcal{M}}\{\ \dot{L}(\ddot{P})-\log\ddot{P}(X^n)\ \}\ ]=$$
$$\sum_{x^n\in\mathcal{X}^n}\bar{P}_{\text{Bayes}}(x^n)(\min_{\ddot{P}\in\mathcal{M}}\{\ \dot{L}(\ddot{P})-\log\ddot{P}(x^n)\ \}\ ). \quad (17.6)$$

We define $\dot{L}_{\text{smml},n}$ to be the code in $\dot{\mathcal{L}}$ that minimizes (17.6). In the sequel we shall simply assume that there is a unique $\dot{L}_{\text{smml},n}$ achieving the minimum in (17.6). Let $\ddot{\mathcal{M}}_{\text{smml},n}$ denote the domain of $\dot{L}_{\text{smml},n}$. $\ddot{\mathcal{M}}_{\text{smml},n}$ is a countable subset of the set $\mathcal{M}$. $\ddot{\mathcal{M}}_{\text{smml},n}$ is the analogue of the discretized parameter set $\ddot{\Theta}_n$ defined in Chapter 10, but it contains distributions rather than parameters. For each individual x^n, the two-part codelength obtained when using the expectation-optimal code $\dot{L}_{\text{smml},n}$,

$$\min_{P\in\ddot{\mathcal{M}}_{\text{smml},n}}\{\dot{L}_{\text{smml},n}(P)-\log P(x^n)\},$$

is achieved for a particular $P\in\ddot{\mathcal{M}}_{\text{smml},n}$. For simplicity, we shall assume that for each x^n there is a unique such P, and denote it by $\ddot{P}_{\text{smml},n}$. Thus, $\ddot{P}_{\text{smml},n}:\mathcal{X}^n\to\ddot{\mathcal{M}}_{\text{smml},n}$ is a function mapping data sequences x^n to corresponding elements of $\mathcal{M}$. The function $\ddot{P}_{\text{smml},n}$ is called the *strict MML (SMML) estimator*. Note that it once again depends on the sample size n. It was introduced in this form by Wallace and Boulton (1975).

It is of some interest to determine a more explicit relation between $\dot{L}_{\text{smml},n}$ and $\ddot{P}_{\text{smml},n}$. For this, note first that the SMML estimator achieves the minimum a priori expected two-part codelength

$$\min_{\ddot{P}_n:\mathcal{X}^n\to\mathcal{M}}\min_{\dot{L}\in\dot{\mathcal{L}}}E_{P\sim W}E_{X^n\sim P}[\dot{L}(\ddot{P}_n)-\log\ddot{P}_n(X^n)]=$$
$$\min_{\ddot{\mathcal{M}}}\min_{\ddot{P}_n:\mathcal{X}^n\to\ddot{\mathcal{M}}}\min_{\dot{L}\in\dot{\mathcal{L}}}E_{X^n\sim\bar{P}_{\text{Bayes}}}[\dot{L}(\ddot{P}_n)-\log\ddot{P}_n(X^n)(X^n)]=$$
$$\min_{\ddot{\mathcal{M}}}\min_{\ddot{P}_n:\mathcal{X}^n\to\ddot{\mathcal{M}}}\{E_{X^n\sim\bar{P}_{\text{Bayes}}}[-\log\ddot{P}_n(X^n)]+\min_{\dot{L}\in\dot{\mathcal{L}}}E_{X^n\sim\bar{P}_{\text{Bayes}}}[\dot{L}(\ddot{P}_n)]\}. \quad (17.7)$$

Here the leftmost minimum in the first line is over all estimators, i.e. functions from samples to elements of $\mathcal{M}$. $\ddot{P}_n(X^n)$ should be read as "the probability of X^n under the distribution to which X^n is mapped by the estimator $\ddot{P}_n$."[10]

10. If $\mathcal{M}$ had been parametric, we could have used the clearer notation $P_{\ddot{\theta}(X^n)}(X^n)$.

The leftmost minimum in the second line is over all countable subsets of $\mathcal{M}$, and the second minimum in the second line is over all "discretized" estimators mapping samples to elements of $\ddot{\mathcal{M}}$. The right-hand side of the final expression in (17.7) shows that the first-stage SMML codelength function $\dot{L}_{\text{smml},n}$ must be equal to the $\dot{L} \in \dot{\mathcal{L}}$ achieving

$$\min_{\dot{L}\in\dot{\mathcal{L}}} E_{X^n\sim\bar{P}_{\text{Bayes}}}[\dot{L}(\ddot{P}_{\text{smml},n})] = \min_{\dot{L}\in\dot{\mathcal{L}}} \sum_{P} \bar{P}_{\text{Bayes}}(\ddot{P}_{\text{smml},n} = P)\cdot \dot{L}(P). \quad (17.8)$$

It now follows by the information inequality that $\dot{L}_{\text{smml},n}$ is given by

$$\dot{L}_{\text{smml},n}(P) = -\log \bar{P}_{\text{Bayes}}(\ddot{P}_{\text{smml},n} = P) = -\log \sum_{x^n:\ddot{P}_{\text{smml},n}=P} \bar{P}_{\text{Bayes}}(x^n). \quad (17.9)$$

Given a model class $\mathcal{M}$, the MML method ideally proceeds by (i) formulating a subjective prior W on $\mathcal{M}$, (ii) determining the corresponding strict MML code $\dot{L}_{\text{smml},n}$ and corresponding strict MML estimator, $\ddot{P}_{\text{smml},n}$, and, (iii) for the given data sequence x^n, compute the value of the corresponding $\ddot{P}_{\text{smml},n}$. It can be seen that this coincides with the P that, among all $P \in \ddot{\mathcal{M}}_{\text{smml},n}$, minimizes the two-part codelength $\dot{L}_{\text{smml},n}(P) - \log P(x^n)$ of the actually given data.

17.4.2 Comparison to MDL

There are three immediate differences with MDL codelength design: first, whereas MDL universal codes need not be two-part, the SMML code is always a two-part code, explicitly encoding a single distribution in $\mathcal{M}$. Second, whereas MDL codes are designed to minimize a worst-case quantity, the SMML code minimizes an *expected* quantity.[11] Third, whereas MDL universal codes seek to minimize luckiness redundancy or regret (relative codelength), the SMML code directly minimizes absolute codelengths. Of these three differences, only the first two are essential. Namely, given the fact that in SMML we take expectations over a prior (the second difference), the third difference disappears.

The Second Difference Makes the Third Disappear To see this, note that the strict MML code achieves

$$\min_{\dot{L}\in\dot{\mathcal{L}}} E_{P\sim W} E_{X^n\sim P}[\min_{\ddot{P}\in\mathcal{M}} (\dot{L}(\ddot{P}) - \log \ddot{P}(X^n))], \quad (17.10)$$

11. Even in expectation-based MDL, one takes the *worst-case* expected regret or redundancy, where the worst-case is over all distributions in the given model. SMML is based on a double expectation instead: the prior-expectation of the expected codelength.

whereas a two-part code minimizing expected (relative) redundancy rather than (absolute) log loss would achieve

$$\min_{\dot{L}\in\dot{\mathcal{L}}} E_{P\sim W} E_{X^n\sim P}[\min_{\ddot{P}\in\mathcal{M}} (\ddot{L}(\ddot{P}) - \log \ddot{P}(X^n))] - [-\log P(X^n)]] = \min_{\dot{L}\in\dot{\mathcal{L}}} E_{P\sim W} E_{X^n\sim P}[\min_{\ddot{P}\in\mathcal{M}} (\dot{L}(\ddot{P}) - \log \ddot{P}(X^n))] - E_{P\sim W}[H(P^{(n)})], \tag{17.11}$$

where $H(P^{(n)})$ is the entropy of the restriction of the source P to the first n outcomes. The last equality, which follows by the linearity of expectation, shows that the $\dot{L}$ achieving the minimum in (17.11) is identical to the $\dot{L}$ achieving the minimum in (17.10), which is just the SMML codelength function $\dot{L}_{\text{smml},n}$. Thus, the two-part code achieving minimum expected codelength is also the code that achieves minimum expected redundancy. It then immediately follows that the two-part estimator minimizing expected two-part codelength (the SMML estimator) is identical to the two-part estimator minimizing expected redundancy. Whereas in the minimax framework, minimizing codelength vs. the redundancy leads to wildly different codes, in the prior expectation framework, the resulting codes coincide.

Philosophical Differences In my view, both the MDL and the MML philosophies are internally consistent, but much more different than is usually thought. This has caused a lot of confusion in debates about the merits of either approach. To give but one example, Rissanen (1989, page 56) writes

> "[Wallace and Freeman] advocate the principle of minimizing the mean codelength [with respect to the prior] ... which strictly speaking does not allow it to be used to select the model class. Indeed, take a model class which assigns the probability $1-\epsilon$ to the string consisting of $0s$ only and the rest equally to all remaining strings. For a small enough ϵ the mean relative to a model can be made as small as we like."

This remark has caused some bewilderment in the MML camp, but from Rissanen's strict M*D*L point of view it makes sense: Rissanen views a prior on hypotheses as a purely pragmatic tool to be used when designing codes. In this book, we have made this view on priors more precise by introducing the more fundamental concept of a "luckiness function" in Chapter 11. The luckiness function indicates how much codelength regret you are willing to accept if your data falls in a certain region. If your data are aligned with the prior/luckiness function you chose, then you will make good inferences already for small samples and you are "lucky." From this point of view, it

makes no sense to design a code that minimizes prior-expected codelength: since you are free to choose the prior, you will always pick a point prior on a low entropy distribution, allowing for an expected codelength approaching 0. Of course, from the subjective Bayesian MML point of view, the prior is seen as something that cannot be chosen at will; it has to seriously reflect one's personal beliefs.

17.4.3 Approximating SMML by the Wallace-Freeman (MML) Estimator

Although (17.7) represents a well-defined optimization problem, in practice, the SMML estimator is very hard to find, and various approximations have been suggested. The most well-known of these is the *Wallace-Freeman estimator* (15.43), also simply known as "(nonstrict) MML estimator." We already defined this estimator in Chapter 15, Section 15.4.3. Because I find the derivation of this estimator in Wallace and Freeman (1987) almost impossible to comprehend, I will give a simplified heuristic derivation based on the two-part M*D*L codes of Chapter 10, Section 10.1. I do this for the special case where $\mathcal{M}$ is a k-dimensional exponential family given in a diffeomorphic parameterization (P_θ, Θ). Note that this is a severe restriction to the general setup, in which $\mathcal{M}$ is often a CUP rather than a parametric model class.

Let W be a prior and let Θ_0 be an arbitrary ineccsi subset of Θ. Theorem 10.1 (see, in particular, (10.4) below the theorem), showed that there exists a 2-part code $\bar{L}_{\text{2-p}}$ such that uniformly for *every* sequence x^n with $\hat{\theta}(x^n) \in \Theta_0$ for n larger than some n_0,

$$\bar{L}_{\text{2-p}}(x^n) \leq -\log \bar{P}_{\text{Bayes}}(x^n) + g(k) + o(1),$$

where $g(k)$ is bounded by $1.05k$ and converges to 0 for large k. Since this holds for every sequence in an arbitrary compact set, we may reasonably conjecture that it also holds in expectation over $\bar{P}_{\text{Bayes}}$. Assuming this is indeed the case, and using the information inequality, it follows that

$$E_{\bar{P}_{\text{Bayes}}}[\bar{L}^{(n)}_{\text{2-p}}(X^n)] \leq E_{\bar{P}_{\text{Bayes}}}[-\log \bar{P}_{\text{Bayes}}(X^n)] + 1.05k + o(1) \leq E_{\bar{P}_{\text{Bayes}}}[\bar{L}^{(n)}_{\text{smml}}(X^n)] + 1.05k + o(1). \quad (17.12)$$

Here $\bar{L}_{\text{smml}}$ represents the codelength function corresponding to the two-part SMML code, i.e. $\bar{L}^{(n)}_{\text{smml}}(x^n) = \dot{L}_n(\ddot{\theta}_{\text{smml},n}) - \log P_{\ddot{\theta}_{\text{smml},n}(x^n)}(x^n)$. Here, for convenience, we switched to the parametric notation of Chapter 10, where we denote distributions P_θ by their parameter θ.

(17.12) shows that *asymptotically* the two-part code of Chapter 10, Section 10.1 is indeed a reasonably good approximation to $\bar{L}_{\text{smml}}$, and in particular that $\bar{L}_{\text{smml}}$ must exhibit the familiar $(k/2 \log n)$ asymptotics shared by both $-\log \bar{P}_{\text{Bayes}}$ and $\bar{L}_{\text{2-p}}$. As a consequence, asymptotically, the two-part estimator $\ddot{\theta}_n$ achieving the minimum codelength in $\bar{L}_{\text{2-p}}$ should behave about as well as the two-part estimator $\ddot{\theta}_{\text{smml},n}$ based on $\bar{L}_{\text{smml}}$.

The essential difference between $\bar{L}_{\text{2-p}}$ and $\bar{L}_{\text{smml}}$ is as follows. $\bar{L}_{\text{2-p}}$ encodes the discretized values $\ddot{\theta} \in \ddot{\Theta}_n$ using the code based on the prior density w, so that $\dot{L}_n(\ddot{\theta}) \approx -\log \int_{\theta \in R(\ddot{\theta})} w(\theta) d\theta$, where $R(\ddot{\theta})$ is the rectangle with center point $\ddot{\theta}$, see Section 10.1. In contrast, from (17.9) we see that $\bar{L}_{\text{smml}}$ encodes $\ddot{\theta}$ based on the probability that some data sequence x^n occurs with $\ddot{\theta}_{\text{smml},n}(x^n) = \ddot{\theta}$.

Wallace and Freeman (1987) were looking, essentially, for an approximation to the SMML estimator that is a continuous function of some statistic of the data, and that would be easy to compute. While the two-part estimator $\ddot{\theta}_n$ is a good approximation of the SMML estimator, it still lacks these two properties. One way to obtain an easily computable continuous estimator is to do a second approximation, and replace $\ddot{\theta}_n$ by its continuum limit. From the proof of Theorem 10.1 ((10.17), page 280), we see that $\ddot{\theta}_n$ is essentially a "discretized MAP estimator," with a prior W_n given by

$$W_n(\ddot{\theta}) \propto \frac{w(\ddot{\theta})}{\sqrt{\det I(\ddot{\theta})}} \left(n^{-k/2} \right), \tag{17.13}$$

where we ignore $(1 + o(1))$-factors. Because for large n, the grid from which the values $\ddot{\theta}$ are taken becomes dense, we see from (17.13) that the continuum limit of the two-part estimator must be given by

$$\hat{\theta}_{\text{wf}}(x^n) := \arg\max_{\theta \in \Theta} \frac{P_\theta(x^n) w(\theta)}{\sqrt{\det I(\theta)}}, \tag{17.14}$$

Here we reason in exactly the same way as we did when showing that the luckiness ML estimator is the continuum limit of the parametric two-part estimator (Chapter 15, Section 15.4.1). (17.14) is just the Wallace-Freeman estimator $\hat{\theta}_{\text{wf}}$ that we defined in Section 15.4.3.

An Apologetic Remark The conference paper (Kontkanen, Myllymäki, Silander, Tirri, and Grünwald 1998) as well as my Ph.D. thesis (Grünwald 1998) contained a theoretical and experimental comparison between MDL and MML approaches. There were two serious mistakes in both works, and in both cases

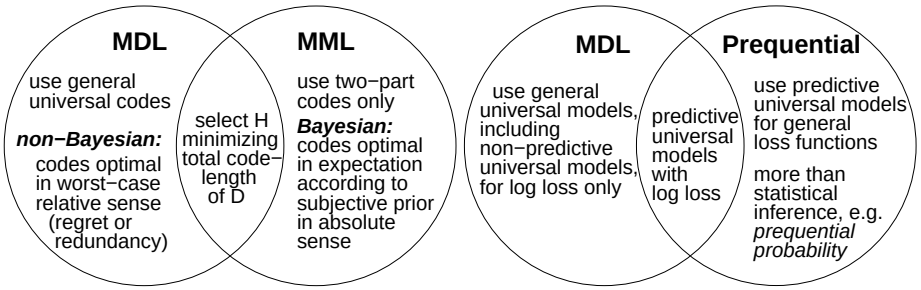

Figure 17.1 Rissanen's MDL, Wallace's MML and Dawid's Prequential Approach

these were caused by myself. First, it was wrongfully claimed that the continuum limit of the two-part code MDL estimator (with uniform luckiness function) would be given by the Bayesian MAP estimator with Jeffreys' prior. This is false: if there is a uniform luckiness function, then the limiting two-part MDL estimator is just the ML estimator, and the ML estimator is not equal to the MAP estimator with Jeffreys prior, except in a parameterization where Jeffreys' prior is uniform; see Section 15.4.1.

Second, while it was correctly claimed that the basic Wallace-Freeman estimator provided silly results when used in combination with the naive Bayes model and small data sets, we did not realize at the time that, in exactly this case, the basic Wallace-Freeman estimator is actually a very bad approximation of the SMML estimator. The latter certainly does provide reasonable estimates. Wallace (2005) explicitly addresses the problems with the Wallace-Freeman approximation for the naive Bayes model. Briefly, even though the naive Bayes model is an exponential family, the sample sizes at which the asymptotics start to play any meaningful role whatsoever are thoroughly unrealistic. Wallace (2005, page 227) shows how the problem can be circumvented by approximating the SMML estimator in a slightly different manner. The problem of unrealistic asymptotics also arises if one replaces the two-part MDL estimator for the naive Bayes model by its continuum limit, the LML estimator.

17.5 MDL and Prequential Analysis

In a series of papers, A.P. Dawid (1984, 1992, 1997) put forward a methodology for probability and statistics based on sequential prediction which he called the *prequential approach*. When applied to model selection problems, it is closely related to MDL: Dawid proposes to construct, for each model $\mathcal{M}^{(j)}$

under consideration, a "probability forecasting system" (a sequential prediction strategy) where the $i+1$-st outcome is predicted based on either the Bayesian posterior $\bar{P}_{\text{Bayes}}(\theta|x^i)$ or on some estimator $\hat{\theta}(x^i)$. Then the model is selected for which the associated sequential prediction strategy minimizes the accumulated prediction error. Related ideas were put forward by Hjorth (1982) under the name *forward validation*.. From Chapter 15, Section 15.2.1 we see that this is just a form of MDL: *every* universal code can be thought of as as prediction strategy, and therefore, in this strict sense, every instance of MDL model selection is also an instance of prequential model selection. It would, however, be strange to call two-part MDL or NML model selection an instance of the prequential approach, since for these methods, in general the horizon n needs to be known in advance; see page 196. Dawid mostly talks about Bayesian and the plug-in universal models for which the horizon does not need to be known, so that the prequential interpretation is much more natural (Chapter 6, Section 6.4). For this reason, I call such codes "prequential" in this book. The terminology is mine: Dawid reserves the term "prequential" for the general framework.

The Infinite-Horizon Prequential Principle From a prequential viewpoint, one may view codes that are neither prequential nor semiprequential as unnatural (Chapter 6, page 196). This may lead one to insist that the only "reasonable" applications of refined MDL are those based on (semi-) prequential codes. One may call this the *infinite-horizon prequential principle*. The terminology is ours; it is different from the weak prequential principle introduced in Section 17.1.2. Note that MDL model selection as defined in Chapter 14 satisfies the infinite-horizon prequential principle, as long as it is based on $\text{CUP}(\text{2-p}, \text{Bayes})$ or $\text{CUP}(\text{2-p}, \text{plug-in})$ codes (the use of a meta-two part code does no harm, since we insisted such codes to be independent of the sample size, which makes them semiprequential). $\text{CUP}(\text{2-p}, \text{2-p})$ codes only satisfy the infinite horizon principle if the two-part code $\bar{L}(\cdot \mid \mathcal{M}_\gamma)$ relative to model $\mathcal{M}_\gamma$ is sample size independent. $\text{CUP}(\text{2-p}, \text{nml})$ codes cannot be used in general – except for the linear regression case where the CNML and NML codes are prequential.

Thus, prequential analysis is usually understood to be "infinite horizon" prequential analysis, and in this sense, it is less general than MDL. On the other hand, Dawid's framework allows for adjusting the sequential prediction loss to be measured in terms of arbitrary loss functions, not just the log loss. In this sense, it is more general than MDL, and is related to the individual sequence prediction literature; see Section 17.9.

There is also a "prequential approach" to probability theory developed by Dawid (Dawid and Vovk 1999) and Shafer and Vovk (2001). Prequential probability and prequential statistics are based on a set of underlying ideas, which one might call the prequential "paradigm" or "philosophy." The prequential philosophy has a lot in common with Rissanen's MDL philosophy, especially the focus on individual sequences rather than ensemble averages. One of its central tenets is the weak prequential principle, which we already introduced in Section 17.1.2. The similarities and difference between MDL and the prequential approach are summarized in Figure 17.1.

MDL and The Weak Prequential Principle The WPP makes eminent sense from the individual-sequence MDL point of view, being reminiscent of Rissanen's tenet that there is no such thing as a true distribution, and "we only have the data," page 28. In a sense it is even more radical, saying that, indeed, when judging the quality of our model, performance of the model on other data than the data at hand may play no role whatsoever. Indeed, MDL inference based on two-part, Bayesian and prequential plug-in universal codes or combinations thereof satisfies the WPP. MDL model selection based on NML codes, however, violates the WPP: since

$$-\log \bar{P}_{\text{nml}}(x^n \mid \mathcal{M}) = -\log P_{\hat{\theta}(x^n)}(x^n) + \log \sum_{y^n \in \mathcal{X}^n} P_{\hat{\theta}(y^n)}(y^n),$$

in order to assess the quality of the model $\mathcal{M}$ (viewed as a prediction strategy with finite horizon $\bar{P}_{\text{nml}}$), one needs to know the distributions $P_{\hat{\theta}(y^n)}(y^n)$ for $y^n \neq x^n$.

Two-Part codes and the WPP Even though sample-size dependent two-part codes, in our terminology, are neither prequential nor semiprequential, they satisfy the WPP. To see this, let $\mathcal{M}$ be some model class, let x^n be the data and suppose we do MDL model or hypothesis selection using a sample-size dependent two-part code. If we consider n to be fixed, we can think of them as defining a sequential prediction strategy with finite horizon (Chapter 6, Section 6.4.1). The log loss of this strategy is the two-part codelength, and to calculate it we only need to know $P(x^n)$ for all $P \in \mathcal{M}$. In particular, we do not need to know $P(y^n)$ for any $y^n \neq x^n$.

Thus, if we want to apply MDL and adhere to the weak prequential principle, we are forced to use two-part codes or (indeed) prequential universal codes such as Bayes and plug-in.

I myself am not sure whether there are any truly undesirable consequences of violating the WPP (except in the situation where $P(y^n)$ for unobserved y^n is

simply unknowable, such as the weather forecasting example), so I have no problems in using (luckiness) NML codes. However, I can sympathize with people who think they should be avoided, and in all cases be replaced by semiprequential universal codes.

17.6 MDL and Cross-Validation

It is well known that one cannot confirm a hypothesis by testing on the data that led one to adopt the hypothesis in the first place. Thus, if Θ represents some complex model with many degrees of freedom, and $-\log P_{\hat{\theta}(x^n)}(x^n)$ is small (achieving a good fit on data x^n), this does not mean anything by itself. To investigate whether Θ is really a good hypothesis for the data, we must either somehow correct for its inherent complexity (as we do in NML model selection), or we must test its behavior on a *distinct* set of data coming from the same source, say $y_1, \ldots, y_m$. This is the rationale for model selection by *cross-validation (CV)* (Stone 1974). In this section we briefly consider CV for selecting between a set of candidate i.i.d. models for the data $D = x^n$. For simplicity, assume n is even. In its simplest form, CV amounts to splitting the data into a *training set* D_1 and a *test set* D_2, both of size $n/2$. D_1 is constructed by randomly selecting $n/2$ elements of D. We then determine the ML estimator $\hat{\theta}(D_1)$ based on the training set and use it to sequentially predict the outcomes in D_2. We record the total prediction error $\hat{\theta}(D_1)$ made on D_2. To make the procedure more robust, we repeat it a few, say M, times, each time making a new random split into training and test set. The M test set prediction errors obtained in this way are then averaged. This procedure is repeated for all models $\mathcal{M}_\gamma$ under consideration. Finally, one selects the model with the smallest average test set prediction errors.

Leave K-out CV In variations of the CV scheme, one may use estimators other than the ML estimator, or loss functions other than log loss. Here we will restrict to (luckiness) ML estimators and log loss. Even with this restriction, the procedure can be substantially varied by changing the relative sizes of training and test set. In *leave-K-out CV*, the size of each test set is set to K outcomes. The simple case we just described corresponds to leave $n/2$-out CV. The consistency and rate of convergence properties of CV strongly depend on whether or not K is allowed to grow with n, and if so, how fast (Shao 1993). For example, in the case we just described, K grows linearly with n. Here we study the case where K remains constant in n, since, as we

now show, this case is most closely related to MDL. The most extreme and most popular version of this case is *leave-one-out cross validation (LOO CV)*. Here, the test set consists of only one outcome. The procedure is usually repeated for all n splits of x^n into a training set with $n-1$ and a test set with 1 outcome. Thus, each model $\mathcal{M}_\gamma$ is associated with its *leave-one-out error*

$$\sum_{i=1}^{n} \left[-\log P_{\hat{\theta}(x^n \setminus x_i)}(x_i) \right], \tag{17.15}$$

where $\hat{\theta}$ is the ML estimator within $\mathcal{M}_\gamma$, and $x^n \setminus x_i$ is the sequence x^n with outcome x_i removed. Model selection proceeds by picking the γ for which (17.15) is minimized. This is obviously related to prequential model selection, as well as to MDL model selection based on the prequential plug-in model, in which we select a model $\mathcal{M}_\gamma$ based on the *accumulated* prediction error

$$\sum_{i=1}^{n} \left[-\log P_{\hat{\theta}(x^{i-1})}(x_i) \right]. \tag{17.16}$$

The main difference is that in MDL, all predictions are done *sequentially*: the future is never used to predict the past.

Shao (1993) shows that LOO CV can be inconsistent for selection between linear models. On the other hand, Li (1987) shows that, under weak conditions, LOO achieves the asymptotically optimal convergence rate for CUP linear models. Thus, in both cases, LOO CV asymptotically behaves like AIC. This was already suggested by Stone (1977), who shows that, under some conditions on the models under consideration, AIC and leave-one-out CV asymptotically select the same model when the number of models under consideration is finite. As we already discussed in Section 17.3.2, the convergence properties of MDL model selection are quite different: it is usually consistent, but not always minimax optimal in terms of the rate.

> One underlying reason for this different behavior seems to be the following. It is clear that, if we let training set and test set fully overlap and the model is complex, then a good fit on the test set (small $-\log P_{\hat{\theta}(y^n)}(y^n)$) is meaningless. This is just the overfitting phenomenon. If the test set partially overlaps with the training set, then the larger the overlap, the less meaningful a good fit on the test set is.
>
> If we do not have additional data available, then we can sequentially test the x_i based on $\hat{\theta}(x^{i-1})$ and add the n resulting prediction errors; this is the prequential idea. Just like with a separate test set, we still have the property that

we can test x_i before we see it. But with leave-one out cross-validation, for $j < i$, x_i is used in the prediction of x_j *and* vice versa. This means that no matter how we order the data, one of the two predictions is made on data that has already been seen and used for other predictions. Thus we cannot maintain that we always test on *unseen data*: in the words of Rissanen (1987), the cross-validation prediction errors are not "honest." Indeed, with a prequential scheme, at the time we predict x_i we have no information at all on how good our prediction for x_i will be; but with a leave-one-out scheme, the prediction errors are correlated: the size of the prediction error for each x_j involves all x_i, $i \neq j$, and therefore does give us some information (albeit admittedly not much) about the prediction errors we will make for x_i with $i \neq j$. Thus, there is some very indirect type of "overlap" between training set and test set after all, which apparently can cause a mild form of overfitting.

17.7 MDL and Maximum Entropy

There are some intriguing connections between MDL and the *Maximum Entropy Principle* ("MaxEnt") for inductive inference, which was first proposed by E.T. Jaynes (1957, 2003). Such connections have been observed by a number of researchers; we mention (Feder 1986; Li and Vitányi 1997; Grünwald 1998) and (Grünwald 2000). Here we follow and extend the observations of (Grünwald 2000). We explain MaxEnt and its relation to minimax codelengths in detail in Chapter 19, Section 19.5. In this section we assume that the reader is familiar with that material. MaxEnt is very frequently applied in practice. To explore the connection to MDL, we need to distinguish between two types of applications. First, MaxEnt can be applied directly on the data; second, it can be applied to select a prior distribution.

1. MaxEnt on the data. This type of application is popular in, for example, computational linguistics (Rosenfeld 1996). Here one has a large sequence of data $x_1, \ldots, x_n$ (for example, a long text written in English), and one records certain statistics of the data (for example, for each pair of words w_1, w_2 appearing in the text, one records the number of times that w_1 is followed by w_2). These statistics are then reformulated as a list of constraints, expressed in terms of a large number of functions $\phi_1, \ldots, \phi_k$, each mapping $\mathcal{X}^*$ to $\mathbb{R}$. For each $j = 1, \ldots, k$, the corresponding constraint is of the form $\sum_{i=1}^{n} \phi_j(x^i) = t_j$. This can be written as one equation in vector form by

defining $\phi = (\phi_1, \ldots, \phi_k)^\top$ and $t = (t_1, \ldots, t_k)^\top$ and writing

$$\sum_{i=1}^{n} \phi(x^i) = t. \tag{17.17}$$

In our natural language example, there would be one ϕ_j for each pair of words w_1, w_2, such that $\phi_j(x^i) = 1$ if $x_{i-1}x_i = w_1w_2$, and $\phi_j(x^i) = 0$ otherwise. Then t_j would be set to the number of times w_1w_2 occurred as a subsequence in x^n. This leads to a long list of constraints (on the data) of form (17.17). In the next step, these are reinterpreted as constraints on the underlying *distribution* P^*, by rewriting them as

$$E_{X^n \sim P^*}\left[\sum_{i=1}^{n} \phi(X^i)\right] = t. \tag{17.18}$$

Such a step may be justifiable if n is large and the functions ϕ only depend on a few of the X_i.

> An important special case arises if for each j, there exists a function $\phi'_j : \mathcal{X} \to \mathbb{R}$ so that $n^{-1}\sum_{i=1}^{n} \phi_j(x^i) = t/n$ can be rewritten as $n^{-1}\sum \phi'_j(x_i) = t/n$. Then (17.18) can be rewritten as $E_{P^*}[\phi'(X)] = t/n$, and the rewriting step amounts to replacing a time average over the data by an ensemble average, which is intuitively reasonable. This makes it plausible that the rewrite (17.18) remains reasonable if ϕ_j does not depend on just one, but only a few of the x_i.

As a "best guess" for the underlying distribution P^*, one now adopts the distribution P_{me} that maximizes the entropy among all distributions that satisfy all given constraints (17.18). As explained in Section 19.5, the maximum entropy will typically be achieved for a distribution $P_{\text{me}} = P_\beta$ that is a member of the exponential family with sufficient statistic $\phi = (\phi_1, \ldots, \phi_k)^\top$. Within this family, the maximum entropy is achieved for the $\hat{\beta}$ that maximizes the likelihood of x^n. Thus, one may reinterpret this form of MaxEnt as consisting of two steps: first, a kind of model "selection," where the set of models to be selected from contains *all* models (sets of probabilistic sources) that can be defined on $\mathcal{X}$. This is followed by maximum likelihood estimation within the chosen model.

We explain in Section 19.5 that P_{me} is the distribution minimizing worst-case expected *absolute* codelength. As explained in Chapter 15, Section 15.4, MDL estimation for a given model $\mathcal{M}$ can be viewed as sequential coding with the goal of minimizing worst-case individual sequence *relative* codelength (regret). There is clearly a relation. To clarify this relation, we first

note that at least if $\mathcal{X}$ is finite, thenP_{me} also has an individual sequence interpretation. Namely, in that case, the *conditional limit theorem* (Csiszár 1984; Cover and Thomas 1991; Grünwald 2001) implies that, under regularity conditions, the maximum entropy distribution is, in a sense, almost identical to the distribution P'_{me} on $\mathcal{X}^n$ that minimizes absolute codelength of individual sequences, in the worst-case over all sequences x^n that satisfy (17.17). Note that P'_{me} is just the uniform distribution on the set of all x^n that satisfy (17.17). With this new insight, we can connect MDL and MaxEnt more closely.

Both MDL estimators[12] relative to a model $\mathcal{M}$ and ME distributions relative to constraints (17.17) may be thought of as (approximations to) codes on $\mathcal{X}^n$. The MDL estimator is the code that tries to achieve codelength $\bar{L}(x^n)$ as close as possible to the shortest expected codelength $L_{\hat{\theta}}(x^n)$ that is obtainable by a code within $\mathcal{M}$, in the worst-case over all sequences on $\mathcal{X}^n$. The maximum entropy distribution is the code that tries to achieve codelength $L_{\text{me}}(x^n)$ as close as possible to the shortest expected codelength $\hat{L}(x^n) = 0$ that is obtainable by a code within the set of all codes on $\mathcal{X}^n$, in the worst-case over all sequences on $\mathcal{X}^n$ that satisfy (17.17).

Thus, in MDL inference, we restrict the class of comparison codes to those that lie within a given model or model class, and we do not restrict the data. In MaxEnt inference, we do not restrict the class of comparison codes at all, so that absolute and relative log loss coincide, but we do restrict the data. Essentially the same story can be told if $\mathcal{X}$ is infinite and the constraints are such that no MaxEnt distribution P_{me} exists. In that case, MaxEnt adherents often assume some "default" or "background" distribution Q on $\mathcal{X}^n$, and adopt the distribution P_{mre} that, among all distributions satisfying (17.17) minimizes the relative entropy relative to Q. Whether such maximum entropy and minimum relative entropy inferences on the data can be justified on external grounds or not seems to depend on the situation one uses them in; see (Grünwald 2000) and (Grünwald and Halpern 2003).

2. MaxEnt on the prior: open problem No. 18. Supposemaximum entropy we are given a model (P_θ, Θ) equipped with some prior density w. Suppose that before we apply this model, we are given some additional information about θ, namely that $\theta \in \Theta_0$ for some convex set $\Theta_0 \subset \Theta$. As an example, Θ may represent the normal family of distributions, and Θ_0 is the subfamily with mean $\mu \geq 4$. How should we update our prior given this information? According to the MaxEnt principle, we should now adopt the prior w' that,

12. For simplicity, we consider MDL estimators restricted to samples of length less than n here.

among all w' satisfying the constraint $\theta \in \mathcal{C}$, is closest to w in relative entropy (KL) distance. This type of MaxEnt application is often interpreted as a tool for objective Bayesian inference (Berger 1985): it tells the Bayesian how to adjust his or her prior given additional information about this prior. Its formal relation to MDL inference has not been investigated. However, if the original prior is a (luckiness-adjusted) Jeffreys' prior, then it seems that the resulting prior will be the luckiness-adjusted prior relative to the set of Θ that satisfy the given constraint. If this is true, then minimum relative entropy inference on the prior would be perfectly consistent with MDL inference. Determining whether something like this is really the case seems an interesting open problem.

17.8 Kolmogorov Complexity and Structure Function; Ideal MDL

Kolmogorov complexity (Li and Vitányi 1997) has played a large but mostly inspirational role in Rissanen's development of MDL. Over the last fifteen years, several "idealized" versions of MDL have been proposed, which are more directly based on Kolmogorov complexity theory. These include extensions of Solomonoff's (1964) original work (Hutter 2003; Hutter 2004), as well as extensions of Kolmogorov's (1965,1974a,1974b) approach (Barron and Cover 1991; Li and Vitányi 1997; Gács, Tromp, and Vitányi 2001; Vereshchagin and Vitányi 2002; Vereshchagin and Vitányi 2004; Vitányi 2005). In both Solomonoff's and Kolmogorov's approaches, hypotheses are described using a universal programming language such as C or PASCAL. Solomonoff's work and its extensions are based on prequential one-part codes, and was discussed in Section 17.2.3 on page 546. Here we very briefly describe Kolmogorov's work and its extensions, which are invariably based on two-part codes. At the end of the section, in Example 17.5, we further investigate the essential difference between coding hypotheses by using a programming language (idealized MDL) and coding hypotheses by implicitly giving their index in some predefined list (two-part MDL as described in this book).

Kolmogorov's Minimum Sufficient Statistic Barron and Cover (1991) describe what is perhaps the most straightforward variation of Kolmogorov's proposal. Given data D, they pick the distribution minimizing

$$K(P) - \log P(D), \tag{17.19}$$

where the minimum is taken over *all* computable probability distributions, and $K(P)$ is the length of the shortest computer program that, when input (x, d), outputs $P(x)$ to d bits precision and halts. While such a procedure is mathematically well-defined, it cannot be used in practice. The reason is that in general, the P minimizing (17.19) cannot be effectively computed. Kolmogorov himself used a variation of (17.19) in which one adopts, among all P with $K(P) - \log P(D) \approx K(D)$, the P with smallest $K(P)$. Here $K(D)$ is the Kolmogorov complexity of D, that is, the length of the shortest computer program that prints D and then halts. This P is known as the *Kolmogorov minimum sufficient statistic*. The resulting method is called the *Kolmogorov structure function* approach (Kolmogorov 1974a,b). As explained by Vitányi (2005), it has several advantages over merely minimizing (17.19). In the structure function approach, the idea of separating data and noise (Section 14.2.1) is taken as basic, and the hypothesis selection procedure is defined in terms of it. The selected hypothesis may now be viewed as capturing all structure inherent in the data; given the hypothesis, the data cannot be distinguished from random noise. Therefore, it may be taken as a basis for *lossy* data compression: rather than sending the whole sequence, we only send the hypothesis representing the "structure" in the data. The receiver can then use this hypothesis to generate "typical" data for it - this data should then look just the same as the original data D. Rissanen views this separation idea as perhaps the most fundamental aspect of "learning by compression." Therefore, in recent work with I. Tabus he has tried to define an analogue of the Kolmogorov structure function for hypotheses that, as in refined MDL, are encoded in a way that is designed to achieve minimax optimal (luckiness) regret. In this way, he connects refined MDL — originally concerned with lossless compression only — to lossy compression, thereby, as he puts it, "opening up a new chapter in the MDL theory" (Vereshchagin and Vitányi 2002; Vitányi 2005; Rissanen and Tabus 2005). Another connection between refined and Kolmogorov-style MDL is due to Poland and Hutter (2005,2006), who consider two-part MDL under the assumption that the data are distributed according to some P in a countable set. They study the predictive properties of two-part MDL estimators, and define variations of two-part estimators with improved prediction quality.

Practical MDL is sometimes seen merely as an approximation to idealized MDL, hampered by the use of less powerful codes to encode hypotheses. The following example shows that the difference is really more subtle than that.

Example 17.5 [Does MDL Allow Cheating?] Suppose we want to do model

selection or universal prediction relative to two singleton models $\mathcal{M}_0 = \{P_0\}$ and $\mathcal{M}_1 = \{P_1\}$. Suppose that P_0 has low, but P_1 has very high Kolmogorov complexity. For concreteness, imagine that P_0 represents a weather forecaster (Example 17.2) who always predicts that the probability that it will rain tomorrow is $1/3$, whereas P_1 is the weather forecaster appearing on Dutch TV, whose predictions are based on a meteorological theory that cannot be implemented by a program of length less than 100 megabytes. From a "practical" MDL perspective, we can design a two-part code relative to $\{P_0, P_1\}$ where both P_0 and P_1 are encoded using just 1 bit. Based on data x^n, we then select the P_j minimizing the two-part codelength, which is just the P_j maximizing the likelihood of the sequence. To some adherents of idealized MDL, this is unacceptable: they argue that instead, we should choose the P_j minimizing $K(P_j) - \log P_j(x^n)$. As a consequence, for small samples, we would never select the complex weather forecaster. This reasoning is incorrect: when we do model selection between two hypotheses such as the weather forecasters, the predictions of the hypotheses can be regarded as *given*, and we can sequentially code data using a "conditional" code (Chapter 3), conditioned on the predictions of the individual hypotheses.

Idealized MDL adherents sometimes dismiss this counterargument on the grounds that practical MDL would then allow for *cheating*: given data x^n, one first looks at the data, then one designs a distribution P_1 that assigns probability 1 to the data x^n, then one performs model selection between $\{P_1\}$ and some other model $\mathcal{M}_0$, using a uniform code on the model index $\{0, 1\}$. In this way one would always choose $\{P_1\}$, now matter how large its Kolmogorov complexity $K(P_1) \approx K(x^n)$ is. Worse, one would even have a large confidence in the result!

Again, the reasoning is not correct: if we design P_1 after seeing data x^n, then P_1 depends on x^n, and therefore $-\log P_1(x^n)$ cannot be interpreted as a codelength function. In the two-stage coding setup, in the sender-receiver interpretation of coding (Chapter 3), this can be seen as follows: when a receiver receives index 1 in the first stage of the code, he cannot decode the remainder of the message, since to decode x^n, he needs to know what P_1 is, and to know what P_1 is, he already needs to know what x^n is.

Concluding, in some cases one can encode hypotheses P_1 with high Kolmogorov complexity using only a few bits, but only if the predictions (probability assignments) of such hypotheses are given in advance (available to both encoder and decoder).

This line of thought does show, however, that there is a crucial difference be-

tween practical MDL on the one hand, and both idealized MDL and purely subjective Bayes on the other hand: suppose that we observe data x^n about an entirely unknown phenomenon, for which we initially have no idea how to model it (see also Example 17.4). In this case, from a practical MDL perspective, it seems a good idea to split the data in two parts, say $x_1, \ldots, x_m$ and $x_{m+1}, \ldots, x_n$, with $m \approx n/2$. Then, based on the first part, one starts thinking and exploring what might be a good model class for the data at hand. Having determined a model class (set of candidate models) $\mathcal{M}$, one proceeds to design a universal code $\bar{L}(\cdot \mid \mathcal{M})$ relative to $\mathcal{M}$, to be used to encode the second part of the data $x_{m+1}, \ldots, x_n$. MDL inference then proceeds as usual, and the confidence in any decision one might make is determined by the amount by which $L(x_{m+1}, \ldots, x_n \mid \mathcal{M})$ compresses data $x_{m+1}, \ldots, x_n$ relative to some null model $\mathcal{M}_0$ (Chapter 14, Example 14.3). Since the model $\mathcal{M}$ is used to encode only a part of the data, the number of bits by which it compresses data relative to $\mathcal{M}_0$ will be smaller than it would have been if the full data $x_1, \ldots, x_n$ had been taken into account. Thus, there will be less confidence in the result of the model selection than there would have been if the full data had been used, but this is how it should be: we used $x_1, \ldots, x_m$ to construct the model $\mathcal{M}$, so then $x_1, \ldots, x_m$ should of course not be used to test the quality of $\mathcal{M}$ as a model for the phenomenon at hand. The first half of the data must be ignored, otherwise cheating would be possible after all. In contrast, in idealized MDL and purely subjective Bayesian approaches, one's prior for the models under consideration is fixed once and for all, before seeing *any* data. Therefore, with these approaches, one can always use the full data sequence $x_1, \ldots, x_n$, and there is a never a need to remove an initial part of it for exploratory purposes. Another way to say this is that the premise "we observe data about which we initially have no idea how to model it" cannot be true from a subjective Bayes or idealized MDL perspective.

17.9 MDL and Individual Sequence Prediction

In its prequential guise, MDL may be viewed as being based on sequential prediction with respect to the logarithmic loss function. In the computational learning and game-theoretic communities, one also studies universal prediction with loss functions other than log loss. Here we compare MDL to this generalized notion of universal prediction; for an overview of the extensive research in this field, see the excellent recent textbook (Cesa-Bianchi and Lugosi 2006).

Suppose one sequentially observes $x_1, x_2, \ldots$, where each $x_i \in \mathcal{X}$. At each point in time, one wants to predict x_i based on the previous data x^{i-1}. The prediction quality is measured by some loss function $\mathbf{L} : \mathcal{X} \times \mathcal{A} \to [0, \infty]$.

Here $\mathcal{A}$ is a space of *actions* or *predictions*. A prediction algorithm is a (computable) function $h : \mathcal{X}^* \to \mathcal{A}$ which maps each initial sequence x^i to an action $h(x^i)$ used to predict x_{i+1}. The loss a prediction algorithm incurs on a sequence x^n is defined to be the sum of the individual losses, i.e. it is given by $\mathbf{L}(x^n, h) := \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1}))$.

If the goal is to compress $x_1, x_2, \ldots,$ then the logarithmic loss function $\mathbf{L}(x, P) = -\log P(x)$ is appropriate. In many other situations, one may be more interested in, say, the squared loss function, where $\mathcal{A} = \mathcal{X} = \mathbb{R}$ and $\mathbf{L}(y, a) = (y-a)^2$, or, if $\mathcal{X}$ is discrete, the $0/1$-loss or *classification loss* function. The latter is defined by setting $\mathcal{A} = \mathcal{X}$, and

$$\mathbf{L}(x, a) = \begin{cases} 1 & \text{if } x \neq a \\ 0 & \text{if } x = a. \end{cases} \tag{17.20}$$

Whereas the log loss is based on probabilistic predictions ($\mathcal{A} = \mathcal{P}$, the set of distributions on $\mathcal{X}$), the squared and the $0/1$-loss correspond to point prediction. As an example of a $0/1$-loss problem, one may think of a weather forecaster who, at each day, only says "it will rain tomorrow" or "it will not rain tomorrow," and one measures her performance by observing how often she predicts correctly.

Let $\mathcal{H}$ be a set of sequential predictors with respect to some known loss function $\mathbf{L}$. The predictors may be hypotheses, but they may also be "experts," such as in the weather forecasting example. Our goal is to design a prediction algorithm $\bar{h}$ that is universal with respect to these predictors for the given loss function $\mathbf{L}$. That is, for all $h \in \mathcal{H}$, the algorithm should satisfy

$$\max_{x^n \in \mathcal{X}^n} \sum_{i=1}^{n} \mathbf{L}(x_i, \bar{h}(x^{i-1})) \leq \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1})) + o(n). \tag{17.21}$$

Entropification It may seem that the type of universal prediction expressed by (17.21) is beyond the scope of MDL approaches, as soon as $\mathbf{L}$ is not the log loss. It turns out, however, that there is a method to "transform" arbitrary sequential prediction problems to log loss sequential prediction problems. Namely, one fixes some $\beta > 0$, say, $\beta = 1$, and, for each action $a \in \mathcal{A}$, one defines the distribution P_a on $\mathcal{X}$ by

$$P_a(x) := \frac{1}{Z(\beta)} e^{-\beta \mathbf{L}(x,a)}, \tag{17.22}$$

where $Z(\beta) = \sum_{x \in \mathcal{X}} e^{-\beta \mathbf{L}(x,a)}$. Note that we implicitly assume here that $Z(\beta)$ does not depend on a. Loss functions for which this holds were called

simple by Grünwald (1998) and Rissanen (2001). Examples of simple loss functions are the $0/1$-loss and other symmetric loss functions such as the squared error loss. Assuming then that $\mathcal{H}$ is a set of predictors to be used against a simple loss function $\mathbf{L}$. We use (17.22) to define, for each $h \in \mathcal{H}$, a corresponding probabilistic source P_h, by

$$P_h(x_i \mid x^{i-1}) := P_{h(x^{i-1})}(x_i) = \frac{1}{Z(\beta)} e^{-\beta \mathbf{L}(x_i, h(x^{i-1}))}. \tag{17.23}$$

Using $P_h(x^n) = \prod P_h(x_i \mid x^{i-1})$ (Chapter 2, page 54), the code L_h corresponding to P_h satisfies

$$L_h(x^n) = -\ln P_h(x^n) = \beta \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1})) + n \ln Z(\beta). \tag{17.24}$$

We see that the codelength (log loss) of x^n under L_h is an *affine (linear plus constant) function of the loss that h makes on x^n, as measured in the loss function* $\mathbf{L}$ *of interest*. Such a transformation of predictors h into probabilistic prediction strategies P_h was suggested by Rissanen (1989), and studied in detail by Grünwald (1998,1999), who called it "entropification." We note that it usually does not make sense to think of P_h as candidates for sources that might have generated the data. They should first and foremost be thought of as log loss prediction strategies or, equivalently, codes. In Section 17.10 we connect entropification to the correspondence between least-squares estimation and ML estimation under the assumptions of Gaussian noise, and we consider what happens if β is allowed to vary, or to be learned from the data.

> It turns out that one can also construct codes L_h with a linear relation to the original loss $\mathbf{L}$ for nonsimple loss functions for which $Z(\beta)$ does depend on $h(x^{i-1})$, but the construction is more complicated (Grünwald 2007).

If we "entropify" each $h \in \mathcal{H}$, we end up with a model of "sources" $\mathcal{M}_{\mathcal{H}} := \{P_h \mid h \in \mathcal{H}\}$, where P_h is given by (17.24). Now suppose we have designed a prequential plug-in universal model $\bar{P}_{\text{plug-in}}$ relative to $\mathcal{M}_{\mathcal{H}}$, that satisfies, for each $P_h \in \mathcal{M}_{\mathcal{H}}$, for all n, x^n,

$$-\ln \bar{P}_{\text{plug-in}}(x^n) \leq -\ln P_h(x^n) + f(n), \tag{17.25}$$

for some slowly growing function $f(n)$, say, $f(n) = O(\ln n)$. Then $\bar{P}_{\text{plug-in}}$ is $f(n)$-universal relative to $\mathcal{M}_{\mathcal{H}}$. Now note that, while until now we used (17.23) to construct sources corresponding to given predictors h, we can also

apply it the other way around: starting from $\bar{P}_{\text{plug-in}}$, we can construct a prediction algorithm $\bar{h}$ such that we have $\bar{P}_{\text{plug-in}} = P_{\bar{h}}$, i.e. for each n, x^n, $P_{\bar{h}}(x^n) = \bar{P}_{\text{plug-in}}(\cdot \mid x^n)$. It then follows immediately from (17.24) and (17.25) that for all $h \in \mathcal{H}$,

$$\sum_{i=1}^{n} \mathbf{L}(x_i, \bar{h}(x^{i-1})) \leq \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1})) + \beta^{-1} f(n),$$

so that $\bar{h}$ is $O(f(n))$-universal relative to $\mathcal{H}$. It seems that we have succeeded to translate arbitrary-loss universal prediction problems to log loss universal prediction, and that MDL is a more general idea than we thought!

The Catch Unfortunately though, there is a catch: it is crucial in the reasoning above that we looked at a *plug-in* universal model $\bar{P}_{\text{plug-in}}$, or equivalently, an "in-model estimator" (Chapter 15, Section 15.2). For many probabilistic model classes $\mathcal{M}$, the best universal models relative to $\mathcal{M}$ are not of the plug-in type. For example, the prediction $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$ corresponding to the Bayesian universal code is a *mixture* of elements of $\mathcal{M}$, which often does not lie itself in $\mathcal{M}$. As already indicated in Chapter 9, for universal prediction in the individual sequence sense, there seems to be an inherent advantage if one is allowed to predict with mixtures. Thus, to get good universal prediction schemes, one would often want to use Bayesian universal models rather than plug-in universal models relative to $\mathcal{M}_{\mathcal{H}}$. But now there is a problem: if the Bayesian prediction $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$ is *not* an element of $\mathcal{M}_{\mathcal{H}}$, there may be no action (prediction) $a \in \mathcal{A}$ such that (17.22) holds, i.e. such that for all $x_{n+1} \in \mathcal{X}$,

$$-\ln \bar{P}(x_{n+1} \mid x^n) = \beta \mathbf{L}(x_{n+1}, a) + \ln Z(\beta). \tag{17.26}$$

In that case, we cannot translate the universal code $\bar{P}$ "back" to a universal prediction algorithm $\bar{h}$ with respect to the original model and loss. At first sight, it seems that the correspondence (17.22) has become useless. In some cases though, entropification can still be useful. Whether or not this is the case, depends on whether the loss function of interest is *mixable*. The concept of mixable loss functions was introduced and developed by V. Vovk and his coworkers in a remarkable series of papers; some highlights are (Vovk 1990; Vovk 2001; Kalnishkan and Vyugin 2002); see also (Littlestone and Warmuth 1994). Roughly speaking, if the loss function is mixable, then a variation of the entropification method can still be used, and the existence of an $f(n)$-universal model for $\mathcal{M}_{\mathcal{H}}$ relative to log loss implies the existence

of an $O(f(n))$ universal model for $\mathcal{H}$ relative to loss function **L**. For example, the squared loss function is mixable as long as it is defined relative to a compact set of outcomes $\mathcal{X} = [-R, R]$ rather than the full real line. Unfortunately, the important $0/1$-loss is *not* mixable. Indeed, if $\mathcal{H}$ consists of a fixed number of N experts, and if we allow the prediction algorithm to randomize (i.e. use a biased coin to determine whether to predict 0 or 1), then the optimal universal $0/1$-loss predictor has worst-case regret (in the worst-case over all types of experts and all sequences x^n) of $\text{ORDER}(\sqrt{n})$, whereas the log loss predictor has a much smaller worst-case regret $\ln N$, independently of n and x^n (Cesa-Bianchi, Freund, Helmbold, Haussler, Schapire, and Warmuth 1997). The latter fact can be seen by noting that the worst-case regret of $\bar{P}_{\text{Bayes}}(\cdot \mid \mathcal{M}_{\mathcal{H}})$ with the uniform prior is bounded by $\ln N$. The upshot is that there exist important nonmixable loss functions **L** such as the $0/1$-loss, which have the property that universal prediction with respect to **L** *cannot* be seen as universal prediction with respect to log loss.[13]

Mixability We now give an informal definition of mixability.[14] As we shall see, mixability cannot be obtained for simple loss functions. Thus, let **L** be a loss function that is not simple, so that $Z(\beta) = Z_a(\beta)$, defined as below (17.23), depends on a. We now define a function $C(\beta) := \sup_{a \in \mathcal{A}} Z_a(\beta)$ and use this to define a defective distribution (Chapter 3, page 94)

$$P_a(x) := \frac{1}{C(\beta)} e^{-\beta \mathbf{L}(x,a)}. \tag{17.27}$$

Now set, for fixed β, $\mathcal{P}_{\mathcal{A}}$ as the set of distributions P_a on $\mathcal{X}$ given by (17.27), so that $\mathcal{P}_{\mathcal{A}}$ contains one distribution for each $a \in \mathcal{A}$. Now let $\overline{\mathcal{P}}_{\mathcal{A}}$ be the convex closure of $\mathcal{P}_{\mathcal{A}}$, i.e. the set of all distributions on $\mathcal{X}$ that can be written as mixtures of elements of $\mathcal{P}_{\mathcal{A}}$.

We say that **L** is mixable if we can choose a $\beta > 0$ such that for *any* mixture $P_{\text{mix}} \in \overline{\mathcal{P}}_{\mathcal{A}}$, there exists an $a \in \mathcal{A}$ such that for all $x \in \mathcal{X}$,

$$-\ln P_{\text{mix}}(x) \geq \beta \mathbf{L}(x, a) + \ln C(\beta). \tag{17.28}$$

Since $P_{\text{mix}}(x) = C(\beta)^{-1} \int e^{-\beta \mathbf{L}(x,a)} w(a) da$ for some prior w on $\mathcal{A}$, (17.28) can be rewritten in the following more common form: for every prior w, there should be an a such that for all x,

$$-\frac{1}{\beta} \ln \int e^{-\beta \mathbf{L}(x,a)} w(a) da \geq L(x, a).$$

13. Nevertheless, some universal predictors that achieve the minimax optimal $0/1$-regret to within a constant, are still based on entropification-related ideas. The important difference is that in such algorithms, the β used in (17.23) varies as a function of n. To get good worst-case performance, one needs to take $\beta = O(1/\sqrt{n})$.

14. Vovk's technical definition is more complicated.

Note that if $\mathbf{L}$ were simple, this would be impossible to achieve since then $C(\beta) = Z(\beta)$ and (17.28) expresses that for all x, $P_a(x) \geq P_{\text{mix}}(x)$, which cannot hold if $P_a(x) \neq P_{\text{mix}}(x)$. Since for nonsimple loss functions, we have $C(\beta) > Z(\beta)$, there sometimes does exist a β for which (17.28) holds after all.

Now define P_h as before, but with $Z(\beta)$ replaced by $C(\beta)$, and for a given set of predictors $\mathcal{H}$, define $\mathcal{M}_{\mathcal{H}} = \{P_h \mid h \in \mathcal{H}\}$. If the mixability condition (17.28) holds, we can modify an $f(n)$-universal code $\bar{P}$ for $\mathcal{M}_{\mathcal{H}}$ into an $O(f(n))$-universal prediction strategy $\bar{h}$ for the loss function $\mathbf{L}$, as long as the predictions $\bar{P}(\cdot \mid x^n)$ can be written as mixtures over the elements of $\mathcal{M}_{\mathcal{H}}$. Thus, unlike in the original entropification approach, we can now also use Bayesian universal codes $\bar{P}_{\text{Bayes}}$. To see this, suppose that $\bar{P}$ is an $f(n)$-universal code for $\mathcal{M}_{\mathcal{H}}$ such that for all n, x^n, $\bar{P}(\cdot \mid x^n) \in \overline{\mathcal{P}}_{\mathcal{A}}$. For each n, x^n, we first set P_{mix} in (17.28) to $\bar{P}(\cdot \mid x^n)$, and then we set $\bar{h}(x^n)$ equal to the a for which (17.28) holds. From (17.28) it is immediate that, for each n, x^n, each $h \in \mathcal{H}$,

$$\beta \sum_{i=1}^{n} \mathbf{L}(x_i, \bar{h}(x^{i-1})) + n \ln C(\beta) \leq -\sum_{i=1}^{n} \ln \bar{P}(x_i \mid x^{i-1}) =$$
$$-\ln \bar{P}(x^n) \leq$$
$$-\ln P_h(x^n) + f(n) \leq \beta \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1})) + n \ln C(\beta) + f(n), \quad (17.29)$$

from which it follows that

$$\sum_{i=1}^{n} \mathbf{L}(x_i, \bar{h}(x^{i-1})) \leq \sum_{i=1}^{n} \mathbf{L}(x_i, h(x^{i-1})) + \beta^{-1} f(n).$$

As an example, if $\mathcal{X} = \{0, 1\}$, $\mathcal{A} = [0, 1]$ and the squared loss is used, then the best achievable β is given by $\beta = 1/2$, and an $f(n)$-universal model relative to $\mathcal{P}_{\mathcal{H}}$ with respect to log loss becomes a $2f(n)$-universal model relative to $\mathcal{H}$ with respect to squared loss. This type of correspondence was initiated by Vovk (1990). Further examples of such correspondences, as well as many other relations between log loss and general universal prediction, are discussed by Yamanishi (1998) in the context of his notion of *extended stochastic complexity*.

MDL Is Not Just Prediction The analysis above suggests that MDL should simply be thought of as the special case of the sequential universal prediction framework, instantiated to log loss, and that all references to data compression may be dropped. This reasoning overlooks three facts. First, Theorem 15.1 tells us that in statistical contexts, there is something special about log loss: in contrast to many other loss functions, with probabilistic predictions, it leads to consistent (prequential) estimators $\bar{P}(\cdot \mid X^n)$. Thus, if a

"true" distribution P^* exists, the KL divergence between $\bar{P}(\cdot \mid X^n)$ and P^* will quickly tend to 0 as n increases. This suggests that, even if we are interested in loss functions $\mathbf{L}$ that are not equal to the log loss, predicting X_{n+1} by $\arg\min_{a \in \mathcal{A}} E_{X_{n+1} \sim \bar{P}(\cdot|X^n)}[\mathbf{L}(X_{n+1}, a)]$ will still be a good idea, at least if n is large enough, at least *if* the data are actually distributed according to a distribution in our model class.

Viewing MDL simply as a special case of universal prediction also ignores the existence of two-part MDL estimation. This form of MDL also has excellent frequentist properties (Theorem 15.3), and is just as fundamental as the other ones. Nevertheless, it is purely based on "nonpredictive" data compression, and interpreting it prequentially is a bit far-fetched. The third problem with the exclusive universal prediction view is model selection. If we compare a finite number of models, and our end goal is to predict future data using a loss function $\mathbf{L}$ that is not equal to the log loss, then it makes perfect sense to judge each model based on accumulated prediction error of an estimator for the model in terms of $\mathbf{L}$ rather than log loss. This is explicitly advocated by the prequential approach, see Section 17.5. It is indeed problematic that MDL does not account for this. However, what if we want to compare an infinite number of models? Then MDL proceeds by adding the codelength needed to encode the models themselves, and we have seen in Chapter 14 that this is sometimes essential. It is not clear how the prequential non-log-loss approach can be extended to deal with this situation.

17.10 MDL and Statistical Learning Theory

Statistical learning theory (Vapnik 1998) is an "agnostic" approach to classification, regression and other conditional prediction problems. One observes a sample $D = (x_1, y_1), \ldots, (x_n, y_n)$ where each $x_i \in \mathcal{X}$ and each $y_i \in \mathcal{Y}$. In regression problems, $\mathcal{Y} = \mathbb{R}$; in classification, $\mathcal{Y}$ is finite; the goal is to match each *feature* X (for example, a bit map of a handwritten digit) with its corresponding *label* or *class* (e.g., a digit); in other words, we want to predict y assuming that x is given. Just as in the individual-sequence prediction framework of the previous section, the prediction quality is measured in terms of a loss function $\mathbf{L} : \mathcal{Y} \times \mathcal{A} \to [0, \infty]$. One usually starts out with a hypothesis class $\mathcal{H}$ consisting of functions $h : \mathcal{X} \to \mathcal{A}$ mapping input values x to corresponding actions a. A standard loss function for regression is the squared loss, $\mathcal{A} = \mathcal{Y}$ and $\mathbf{L}(y, a) = (y - a)^2$. A standard loss for classification is the $0/1$-loss, defined as in (17.20), with $\mathcal{X}$ replaced by $\mathcal{Y}$. Based on a sample D,

one wants to learn a $h \in \mathcal{H}$ that makes good predictions on future data. In learning theory, this is formalized by assuming that data (X_i, Y_i) are jointly i.i.d. according to some unknown source P^*. In the basic framework, one makes *no assumptions at all about* P^*, except that data are *i.i.d.* according to P^*. Thus, just like in MDL, one takes an "agnostic" stance, but this is realized in a totally different way.

We may view P^* as a joint distribution on $\mathcal{X} \times \mathcal{Y}$ and define the *risk* of a hypothesis h as $E_{X,Y\sim P^*}[\mathbf{L}(Y, h(X))]$. In the machine learning community, the risk is often called "error function," and the risk of h is called the *generalization error* of h. By the law of large numbers, a hypothesis with small generalization error will, with very high probability, make good predictions on future data from P^*. For a given hypothesis class $\mathcal{H}$, we further define

$$\mathbf{L}^* := \inf_{h\in\mathcal{H}} E_{X,Y\sim P^*}[\mathbf{L}(Y, h(X))]. \tag{17.30}$$

$\mathbf{L}^*$ is the best expected loss that can be obtained by a hypothesis in $\mathcal{H}$. Thus, the goal of learning theory can now be rephrased as: find a good learning algorithm mapping, for each n, data $(x_1, y_1), \ldots, (x_n, y_n)$ to hypotheses $\dot{h}_n \in \mathcal{H}$ such that $E_{P^*}[\mathbf{L}(Y, \dot{h}_n(X))] \to \mathbf{L}^*$, *no matter what the distribution* P^* *is*. Here the convergence may be in P^*-expectation, or with high P^*-probability. If $\mathcal{H}$ is sufficiently simple, then this goal can be achieved by *empirical risk minimization* (ERM): for sample $(x_1, y_1), \ldots, (x_n, y_n)$, we simply pick any $\hat{h} \in \mathcal{H}_\gamma$ minimizing the *empirical risk* $n^{-1}\sum_{i=1}^n \mathbf{L}(Y_i, h(X_i))$. This is just the $\hat{h}$ that achieves the smallest error on the sample.

Example 17.6 [Polynomial Regression] Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ and let $\mathcal{H}_\gamma$ be the set of polynomials of degree γ. For simplicity, assume that $\gamma = 2$. Suppose we are interested in predicting y given x against the square loss. Given points $(x_1, y_1), \ldots, (x_n, y_n)$, ERM tells us to pick the polynomial $\hat{h}_n$ that achieves the optimal least-squares fit $\min_{h\in\mathcal{H}_\gamma} \sum (y_i - h(x_i))^2$. The optimal polynomial $\tilde{h} \in \mathcal{H}$ is the polynomial achieving $\min_{h\in\mathcal{H}_\gamma} E_{P^*}[(Y - h(X))^2] = L^*$. By the *uniform law of large numbers* (Vapnik 1998), it holds that $\hat{h}_n \to \tilde{h}$, in the sense that

$$E_{P^*}[(Y - \hat{h}_n(X))^2] \to L^*.$$

in P^*-probability and in P^*-expectation. This holds no matter what P^* is, so that ERM can be used to learn a good approximation of $\tilde{h}$, no matter what P^* is. P^* may of course be such that even the optimal $\tilde{h} \in \mathcal{H}$ predicts Y quite badly. This happens, for example, if $P^*(Y \mid X)$ is essentially flat (has fat tails) and does not depend on X. So we cannot always guarantee that, based on a

small sample, we will learn, with high probability, a $\hat{h}_n$ which predicts well. We can guarantee however that, based on a small sample, we will learn a $\hat{h}_n$ which predicts almost as well as the best predictor $\tilde{h} \in \mathcal{H}$.

Now consider the model $\mathcal{P}^{(\gamma+1)}$ (Example 2.9, page 64). This is the linear model corresponding to $\mathcal{H}^\gamma$, i.e. it assumes that Y_i are independent given X_i, and normally distributed with mean $h(X_i)$ for some $h \in \mathcal{H}^\gamma$, and some fixed variance σ^2. As we pointed out in Chapter 12, Section 12.3, least-squares estimation for $\mathcal{H}_\gamma$, which is what ERM amounts to in this case, corresponds to ML estimation for $\mathcal{P}^{(\gamma+1)}$. Yet there is an important interpretation difference: rather than modeling the noise as being normally distributed, ERM seeks to learn functions $h \in \mathcal{H}_\gamma$ in a way that leads to good predictions of future data with respect to the squared loss, *even if P^* is such that the noise is very different from the normal distribution, i.e. even if $P^*(Y|X)$ is not normal at all*. Implementing this goal leads to algorithms that differ significantly with MDL and Bayes once we consider larger classes of hypotheses such as the set of all polynomials considered. This is the topic of the next subsection.

17.10.1 Structural Risk Minimization

If $\mathcal{H}$ contains predictors of arbitrary complexity, then ERM will fail. For example, this happens if $\mathcal{H}$ is the set of all polynomials of each degree. Then for a sample of size n, ERM will tend to select a polynomial of degree $n-1$ that perfectly fits the data. As we already saw in Chapter 1, Example 1.3, such a polynomial will severely overfit the data and will not lead to good generalization performance. This is of course analogous to the maximum likelihood estimator for the linear model defined relative to the set of all polynomials, which also corresponds to a polynomial of degree $n-1$. For this situation, Vapnik (1982,1998) proposed the *structural risk minimization (SRM)* method; see also (Bartlett, Boucheron, and Lugosi 2001). The idea is to carve up a hypothesis class $\mathcal{H}$ into subsets $\mathcal{H}_1, \mathcal{H}_2, \ldots$ such that $\bigcup_\gamma \mathcal{H}_\gamma = \mathcal{H}$. The subclasses $\mathcal{H}_\gamma$ are typically nested, $\mathcal{H}_\gamma \subset \mathcal{H}_{\gamma+1}$, and correspond to what we call "models" in this book. In our polynomial example, $\mathcal{H}_\gamma$ would be the set of polynomials of degree γ. The idea is to first select a model $\mathcal{H}_{\dot{\gamma}}$ for the given data $(x_1, y_1), \ldots, (x_n, y_n)$ by minimizing some tradeoff between the complexity of $\mathcal{H}_\gamma$ and if the fit of $\hat{h}_\gamma$, the best-fitting predictor within $\mathcal{H}_\gamma$. In the simplest forms of SRM, this tradeoff is realized by picking the $\dot{\gamma}_n$ achieving

$$\min_{\gamma \in \Gamma} \quad f_n(\hat{\mathbf{L}}_\gamma, \mathrm{COMP}_{\mathrm{srm}}(\mathcal{H}_\gamma)). \tag{17.31}$$

Here

$$\hat{\mathbf{L}}_\gamma := \inf_{h \in \mathcal{H}_\gamma} \frac{1}{n} \sum_{i=1}^{n} \mathbf{L}(y_i, h(x_i))$$

measures the empirical error that is achieved by the $h \in \mathcal{H}_\gamma$ that minimizes this empirical error; note that this is analogous to the quantity $-\log P_{\hat{\theta}(x^n)}(x^n)$ appearing in MDL complexity tradeoffs. $f_n : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is some function that is increasing in both arguments. $\text{COMP}_{\text{srm}}(\mathcal{H}_\gamma)$ is some function that measures the complexity of the set of predictors $\mathcal{H}_\gamma$. Intuitively, the more patterns for which there is an $h \in \mathcal{H}_\gamma$ that fits them well, the larger the complexity. As we show below, the complexity measures used in SRM are often directly or indirectly related to the number of bits needed to describe an element of $\mathcal{H}_\gamma$ using a worst-case optimal code. Thus, (17.31) is reminiscent of two-part code MDL: we pick the $\mathcal{H}_\gamma$ optimizing a tradeoff between loss on the data and hypothesis class complexity. Different forms of SRM use different definitions of COMP_{srm}; some of these definitions are *data-dependent*, in the sense that $\text{COMP}_{\text{srm}}(\mathcal{H}_\gamma)$ is really a function not just of $\mathcal{H}_\gamma$ but also of x^n. Again this is reminiscent of MDL, where the parametric complexity of regression models also depends on the design matrix $\mathbf{X}$.

In two-part MDL, both the loss and the complexity are measured in bits, and they are simply added to one another. In SRM, the loss and the complexity are measured in different units, and rather than just adding them, the tradeoff is in terms of a more complicated function f_n which depends on the sample size n, and which increases both if the empirical loss $\hat{L}$ and if the complexity $\text{COMP}_{\text{srm}}(\mathcal{H}_\gamma)$ increase. We postpone giving explicit examples of f_n until the next subsection, where we discuss a variation of SRM that is more closely related to MDL. The tradeoff (17.31), the complexity measures COMP_{srm} and the function f_n are all designed so as to make sure that $\hat{h}_{\hat{\gamma}}$ converges to the best hypothesis in $\mathcal{H}$ as fast as possible, in the sense that

$$E_{P^*}[\mathbf{L}(Y, \hat{h}_{\hat{\gamma}_n})] \to \mathbf{L}^*,$$

with high P^*-probability, or in P^*-expectation. Here $\mathbf{L}^*$ is defined as in (17.30). Again, this will be explained in detail in Section 17.10.2, where we give an explicit example. The "best possible" choices for COMP_{srm} may depend on the hypothesis class $\mathcal{H}$ under consideration.

Complexity Measures for Classification We now give some examples of complexity measures COMP_{srm} that have been used in the SRM literature. We concentrate on classification settings with $\mathcal{Y} = \{0, 1\}$ and the $0/1$-loss

function as defined by (17.20). This is the type of SRM application that has most often been studied in practice.

For a given sample x^n, we may partition any given $\mathcal{H}$ into N equivalence classes $\{\mathcal{H}_1, \ldots \mathcal{H}_N\}$, where hypotheses fall into the same class $\mathcal{H}_j$ if and only if they agree on all given x_i. That is, for all $j \in \{1, \ldots, N\}$, for all $h, h' \in \mathcal{H}_j$, for all x_i with $1 \leq i \leq n$, $h(x_i) = h'(x_i)$; and for each $h \in \mathcal{H}_j, h' \in \mathcal{H} \setminus \mathcal{H}_j$, there is an x_i with $h(x_i) \neq h'(x_i)$. $N = N(x^n)$ depends on the input data x^n, and must satisfy $N(x^n) \leq 2^n$, since y^n can only take on 2^n distinct values. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{H}$ is defined as the largest n for which there exists a sample x^n with $N(x^n) = 2^n$ (Vapnik and Chervonenkis 1971). This is the largest n for which there exists a sample that can be classified in all 2^n possible ways by elements of $\mathcal{H}$. Clearly, this measures something like the "richness" of $\mathcal{H}$. The VC dimension was historically the first complexity notion used in SRM approaches. These were based on applications of (17.31) with $\mathrm{COMP}_{\mathrm{srm}}(\mathcal{H})$ instantiated to the VC dimension of $\mathcal{H}$.

According to *Sauer's lemma* (Vapnik and Chervonenkis 1971), if $\mathcal{H}$ has VC-dimension d, then for all n, x^n, $N(x^n)$ is bounded by $\sum_{j=0}^{d} \binom{n}{j}$, so that, for $n > 1$, $N(x^n) \leq n^d$. Thus, suppose there is a $h \in \mathcal{H}$ that fits the data perfectly, i.e. $h(x_i) = y_i$ for $i = 1, \ldots, n$. Then $h \in \mathcal{H}_j$ for some j, and in order to encode y^n given x^n and hypothesis class $\mathcal{H}$, it suffices to describe the number j. This takes at most $d \log n$ bits, since we must have $j \in \{1, \ldots, N(x^n)\}$ and by Sauer's lemma, $N(x^n) \leq n^d$.

More Relations between Complexities in Learning Theory and in MDL Interestingly, the VC-dimension was originally introduced to provide a distribution-independent upper bound for what Vapnik calls the *annealed entropy*, defined, for given $\mathcal{H}$, as $\log E_{X^n \sim P^*}[N(X^n)]$. This quantity cannot be computed directly because it depends on the unknown distribution P^*, but by Sauer's lemma, it is bounded, for all $n > 1$, by $d \log n$, where d is the VC-dimension of $\mathcal{H}$. Most other authors call the annealed entropy simply "entropy," which I think is less fortunate because, unlike the Shannon entropy, it does not have a direct expected codelength interpretation. However, it would have such an interpretation if we exchanged $\log$ and expectation, just like Rényi entropy (no direct coding interpretation) becomes equal to Shannon entropy (direct coding interpretation) if we exchange $\log$ and expectation.

Open Problem No. 19: Rademacher vs. Parametric Complexity The *empirical Rademacher complexity* (Bartlett, Boucheron, and Lugosi 2001; Boucheron, Bousquet, and Lugosi 2005) is a more recent complexity notion used in SRM approaches. It bears a resemblance to the parametric complexity $\mathrm{COMP}^{(n)}(\mathcal{M})$

although it is unclear whether it can be given a coding interpretation. The empirical Rademacher complexity is used in classification problems where $\mathcal{Y} \in \{-1, 1\}$, and $\mathcal{H}$ consists of real-valued predictors $h : \mathcal{X} \to \mathbb{R}$. Many classification models used in practice employ such h to predict y against the $0/1$-loss, and then $h(x) > 0$ is interpreted as a prediction of 1, and $h(x) < 0$ is interpreted as a prediction of -1. This is the case in, for example, feedforward neural networks and in support vector machines (SVMs; see (Schölkopf and Smola 2002)). To simplify definitions, we will assume that for each $h \in \mathcal{H}$, there is a $h' \in \mathcal{H}$ such that for all x, $h(x) = -h'(x)$. This condition is satisfied for SVMs and feedforward neural networks. In such cases, the empirical Rademacher complexity of $\mathcal{H}$, relative to inputs $x_1, \ldots, x_n$, is defined as

$$\hat{R}^{(n)}(\mathcal{H}) := n^{-1} 2^{-n+1} \sum_{y^n \in \{-1,1\}} \sup_{h \in \mathcal{H}} \sum_{i=1}^{n} y_i h(x_i)$$

Just like $\mathrm{COMP}^{(n)}(\mathcal{M})$ in regression problems (Chapter 14, Section 14.5), this quantity is data-dependent: it depends on the input values $x_1, \ldots, x_n$. Let us compare it more closely to the parametric complexity of an i.i.d. conditional probabilistic model $\mathcal{M} = \{P_\theta(Y \mid X) \mid \Theta\}$ for $\mathcal{Y}$, which, for given x^n, may be written (Chapter 7) as

$$\mathrm{COMP}^{(n)}(\mathcal{M}) = \log \sum_{y^n \in \{-1,1\}} \sup_{P \in \mathcal{M}} e^{-\sum_{i=1}^{n} [-\log P(y_i | x_i)]}.$$

In both cases, one takes the sum over all possible realizations of the data y^n, of the best fit that can be achieved for that particular y^n. In the MDL case, the fit is measured in terms of the exponent of minus log loss . In the structural risk minimization case, the fit is measured in terms of a "smoothed" version of the $0/1$-loss.

A final connection between complexity notions in learning theory for classification and MDL is provided by the so-called *compression schemes* (Floyd and Warmuth 1995). Here, one focuses on hypothesis classes $\mathcal{H}$ such that each $h \in \mathcal{H}$ can be uniquely identified by a finite number of input values x_i. For example, we may have $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y}$ is the class of 'rectangles' on $\mathcal{X}$. That is, each $h \in \mathcal{H}$ has $h(x) = 1$ if and only if x falls in some rectangle with sides running parallel to the axis of $\mathcal{X}$. Then each h may be identified by two points in the plane (its lower left and upper right corner). Given a sample of input points $x_1, \ldots, x_n$, one can now "encode" a hypothesis h by giving the indices (j_1, j_2) of two of the x_i-points, which are interpreted as the lower left and upper right corner of h. Thus, one needs $\log \binom{n}{2}$ bits to encode a rectangle. Such a method of representing h is called a compression scheme; the complexity of a class $\mathcal{H}$ may be measured by the number of x-values that must be provided in order to identify an element of h uniquely.

Such relations between learning and coding complexity notions notwithstanding, there is usually no direct interpretation of (17.31) in terms of minimizing a codelength. This is due to the fact that the function f_n depends on n, $\hat{\mathbf{L}}_\gamma$ and $\mathrm{COMP}_{\mathrm{srm}}$ in a complicated manner. Below we clarify this issue in the context of the PAC-Bayesian approach to learning theory, a variation of SRM which has complexity penalties that resemble those of MDL even more closely.

17.10.2 PAC-Bayesian Approaches

In the PAC-Bayesian method of McAllester (1998,1999,2002), complexity penalties are measured by a user-supplied prior distribution W, or equivalently, a codelength function L, on *hypotheses* $\mathcal{H}$. Although this prior distribution may be chosen subjectively, its interpretation is quite different from that of a subjective prior in Bayesian statistics. It is (much) more closely related to MDL's luckiness interpretation of codelength functions.[15] For our purposes, it is sufficient to discuss a simplified version of the method, with a level of sophistication inbetween that of the so-called "Occam's Razor bound" (a precursor to PAC-Bayes due to Blumer, Ehrenfeucht, Haussler, and Warmuth (1987)) and the PAC-Bayes method itself. Below we describe this simplification and its rationale, highlighting similarities and differences with MDL. For simplicity we restrict to hypothesis selection in a classification setting, $\mathcal{Y} = \{0, 1\}$ with the $0/1$-loss function, and a countable set of hypotheses $\mathcal{H}$. The set of hypotheses may be arbitrarily complex though, in the sense of having infinite VC-dimension; for example, $\mathcal{H}$ may be the set of all decision trees with an arbitrary depth and arbitrary number of leaves, and with decision functions based on rational numbers. See (McAllester 2003) for extensions to uncountable hypothesis classes, stochastic hypothesis "averaging," and other loss functions.

Simplified PAC-Bayes Hypothesis Selection Let $\mathbf{L}$ be the $0/1$-loss function. In the remainder of this section, we abbreviate $n^{-1} \sum_{i=1}^{n} \mathbf{L}(Y_i, h(X_i))$ to $\mathbf{e}_{\mathrm{emp}}(h)$, and $E_{X,Y \sim P^*}[\mathbf{L}(Y, h(X))]$ to $\mathbf{e}(h)$.

In order to apply the PAC-Bayesian method, we must first fix some *confidence level* δ, the meaning of which will become clear later. For concreteness, we could take $\delta = 0.05$. We could also choose δ as a function of the sample

15. Indeed, some of the bounds on which PAC-Bayesian model selection and averaging are based have been called *PAC-MDL* bounds in the literature (Blum and Langford 2003).

size n, say, $\delta = 1/n$. With this choice, the influence of δ becomes almost negligible for large n. We must also fix a "prior" W on the countable set $\mathcal{H}$, and define the codelength function (measured in nats) $L(h) = -\ln W(h)$. Now suppose we are given data $(x_1, y_1), \ldots, (x_n, y_n)$. Then according to simplified PAC-Bayes, we should pick the hypothesis $\ddot{h}$ minimizing, over all $h \in \mathcal{H}$,

$$n\mathbf{e}_{\text{emp}}(h) + 2L(h) + \sqrt{n} \cdot \sqrt{\mathbf{e}_{\text{emp}}(h)(8L(h) - \ln \delta)}. \tag{17.32}$$

Why would this be a good idea? The hypothesis selection rule (17.32) is based on a *generalization bound* expressed in Proposition 17.1 below. As we will see below, the best performance guarantee on the generalization error $\mathbf{e}(h)$ given by that bound is achieved for the $\ddot{h}$ minimizing (17.32). This is a typical instance of what we called the *frequentist design principle* and criticized in Section 17.1.1: one proves a certain frequentist property of sets of classifiers, and then one designs a hypothesis selection algorithm that is optimal relative to the proven property. In the case of PAC-Bayes and other statistical learning methods, I have not much objections against this principle, since the sole assumption on which it is based is that the data are i.i.d. Indeed, this is one of the few examples of a modeling assumption which may actually be quite realistic in some situations.

Proposition 17.1 Let $\mathcal{H}$ be an arbitrary countable set of classifiers. Assume $(X_1, Y_1), \ldots, (X_n, Y_n)$ are i.i.d. P^*. Then no matter what P^* is, for all $h \in \mathcal{H}$, in particular, for the $\ddot{h}$ minimizing (17.32), we have, with P^*-probability at least $1 - \delta$, that

$$n\mathbf{e}(h) \leq n\mathbf{e}_{\text{emp}}(h) + 2L(h) - \ln \delta + \sqrt{n} \cdot \sqrt{8\mathbf{e}_{\text{emp}}(h)(L(h) - \ln \delta)}.$$

Results of this type, but with cruder notions of complexity, were originally called *probably approximately correct (PAC)* generalization bounds. This explains the term "PAC-Bayes." The bound says that, simultaneously for *all* hypotheses $h \in \mathcal{H}$, their generalization error is not much larger than their error on the training set plus a slack, which depends on the prior on h: the bound holds for all h at the same time, but is stronger for h with a large "prior" $W(h) := e^{-L(h)}$. This means that, for each hypothesis h, with high P^*-probability, if $L(h)$ is small, then its performance on a future test set is not much worse than its performance on the training set. Since our goal is to find a hypothesis with small generalization error $\mathbf{e}(h)$, it may be a good idea to pick the h for which Proposition 17.1 provides the smallest *upper bound* on the generalization error. This is exactly the h we pick in the PAC-Bayesian

method. Note that there is an analogy to our luckiness approach: if there exists a h with small empirical error $\mathbf{e}_{\text{emp}}(h)$ and large "prior" $W(h)$, then we were *lucky* and get a good (small) upper bound on future performance. If there exists no such h, than we are not lucky, but, by Proposition 17.1, we *know* in that case that the h chosen in (17.32) may predict badly in the future. The main difference to our notion of "luckiness" is that here it refers to generalization error for future data (an expected quantity), whereas in individual-sequence MDL, it refers to individual sequence codelength of the given data; although we do use it as an indication of how much confidence we have in the prediction quality (codelength) that we achieve on future data.

Proof: For each $h \in \mathcal{H}$, let $Z_{h,i}:=|Y_i - h(X_i)|$. Then $Z_{h,1}, Z_{h,2}, \ldots$ are i.i.d. Bernoulli distributed, with mean $\mu_h^* = P^*(Z_{h,1} = 1)$. Let $\hat{\mu}_h$ be the corresponding ML estimator based on data Z^n. It follows from Theorem 19.2 in Chapter 19 that, for all $K > 0$,

$$P^*(n\mathrm{D}(\hat{\mu}_h \| \mu_h^*) \geq K) \leq e^{-K}.$$

Therefore,

$$P^*(\exists h \in \mathcal{H} \,:\, n\mathrm{D}(\hat{\mu}_h \| \mu_h^*) \geq L(h) - \ln \delta) \leq \sum_{h \in \mathcal{H}} P^*(n\mathrm{D}(\hat{\mu}_h \| \mu_h^*) \geq L(h) - \ln \delta) \leq \sum_{h \in \mathcal{H}} e^{-L(h)} e^{\ln \delta} \leq \delta. \quad (17.33)$$

where the first inequality is the union bound, the final inequality is Kraft's, and we set $K = L(h) - \ln \delta$. Now if $Z_1, \ldots, Z_n$ are i.i.d. Bernoulli with mean μ^* and $\hat{\mu}$ represents the ML estimator, then we must have

$$\mathrm{D}(\hat{\mu} \| \mu^*) \geq \frac{(\hat{\mu} - \mu^*)^2}{2\mu^*}.$$

This follows by a Taylor approximation of the type we performed in Chapter 4, Section 4.3, using the fact that the Fisher information is given by $I(\mu^*) = 1/(\mu^*(1-\mu^*))$. Using the fact that $\mu_h^* = \mathbf{e}(h)$ and $\hat{\mu}_h = \mathbf{e}_{\text{emp}}(h)$, together with (17.33), taking square roots, and rearranging terms, we get

$$P^*\left(\exists h \in \mathcal{H} \,:\, \frac{|\mathbf{e}(h) - \mathbf{e}_{\text{emp}}(h)|}{\sqrt{\mathbf{e}(h)}} \geq \sqrt{\frac{2(L(h) - \ln \delta)}{n}}\right) \leq \delta. \quad (17.34)$$

If $\mathbf{e}_{\text{emp}}(h) < \mathbf{e}(h)$, then $\mathbf{e}_{\text{emp}}(h)/\sqrt{\mathbf{e}(h)} < \sqrt{\mathbf{e}_{\text{emp}}(h)}$. With this observation, (17.34) implies that

$$P^*\left(\exists h \in \mathcal{H} \,:\, \sqrt{\mathbf{e}(h)} - \sqrt{\mathbf{e}_{\text{emp}}(h)} \geq \sqrt{\frac{2(L(h) - \ln \delta)}{n}}\right) \leq \delta.$$

The result now follows by moving $\sqrt{\mathbf{e}_{\text{emp}}(h)}$ to the right of the inequality inside the probability, and squaring both sides inside the probability. □

Relation to SRM The overall strategy to arrive at the hypothesis selection criterion (17.32) was to first derive a uniform generalization bound, relating empirical error to generalization error, that holds for *all* $h \in \mathcal{H}$ simultaneously. This bound then motivates an algorithm that selects the h which, for the given data, gives, with high probability, the smallest upper bound on generalization error. The SRM method which we described further above is invariably based on exactly the same idea: one first proves a generalization bound which holds for all $h \in \mathcal{H}$ simultaneously; this bound may depend on complexity notions such as VC-dimension applied to subclasses $\mathcal{H}_\gamma \subset \mathcal{H}$. One then designs an algorithm that selects the $\mathcal{H}_\gamma$ containing the h for which the bound is optimal.

17.10.3 PAC-Bayes and MDL

MDL and learning theory approaches may seem to be very different: in the former, hypotheses are probability models; in the latter, they are (sets of) predictors relative to arbitrary loss functions (most often, the $0/1$-loss). In the former, no probabilistic assumptions are made; in the latter, it is assumed that data are sampled from an i.i.d., but otherwise arbitrary, unknown source.

The first difference is less essential than it seems: the probabilistic sources appearing in MDL are first and foremost interpreted not as probability distributions but rather as *codes* or equivalently, predictors relative to the log loss function. Just as we did for the individual sequence prediction (Section 17.9), we may "entropify" any arbitrary hypothesis class $\mathcal{H}$ together with a loss function $\mathbf{L}$, so that it becomes a model class $\mathcal{P}_\mathcal{H}$ consisting of conditional i.i.d. sources such that for some $\beta > 0$, for each $h \in \mathcal{H}$, there is exactly one $p_h \in \mathcal{P}_\mathcal{H}$ satisfying, for all n, x^n, y^n,

$$-\ln P_h(y^n \mid x^n) = \beta \sum_{i=1}^{n} \mathbf{L}(y_i, h(x_i)) + n \ln Z(\beta). \tag{17.35}$$

Thus, the log loss that P_h achieves on any sequence of data is a fixed affine (linear plus constant) function of the loss achieved by h as measured in the original loss function $\mathbf{L}$. To construct P_h, we use an analogue of (17.23):

$$P_h(x_i \mid y_i) := \frac{1}{Z(\beta)} e^{-\beta \mathbf{L}(x_i, h(y_i))}, \tag{17.36}$$

where $Z(\beta) = \sum_{x \in \mathcal{X}} e^{-\beta \mathbf{L}(x, h(y_i))}$. P_h is extended to a conditional source by taking product distributions. Note that P_h is an i.i.d. conditional source here, whereas in the individual sequence prediction, $P_h(x_i \mid x^{i-1})$ strongly depended on x^{i-1}. The difference arises because in the present setup, the predictors h only use side information x_i rather than information about the past. In the case where $\mathcal{H}$ is a set of functions from $\mathcal{X}$ to $\mathbb{R}$ and $\mathbf{L}$ is the squared error loss function, $\mathcal{P}_{\mathcal{H}}$ is just the linear regression model with Gaussian noise with fixed variance $\sigma^2 = 1/2\beta^{-1}$.

Now suppose we have a countable set of classifiers $\mathcal{H}$. Equation (17.35) suggests the following MDL approach to hypothesis selection for classification: first, we fix some $\beta > 0$ and we transform each $h \in \mathcal{H}$ into the corresponding source (or, more appropriately, log loss prediction strategy) P_h. Second, we perform two-part code MDL on the resulting model $\mathcal{P}_{\mathcal{H}}$, using some code $L(h)$ for encoding hypotheses in $\mathcal{H}$ (Of course, we may use minimax and luckiness principles to guide our choice of $L(h)$, but the details of this choice do not matter below). Using the abbreviations for $0/1$-loss introduced above, this amounts to selecting the $\ddot{h}$ minimizing the two-part codelength

$$n\beta \mathbf{e}_{\text{emp}}(h) + n \ln Z(\beta) + L(h).$$

This is closely related to, but much simpler than, the PAC-Bayes hypothesis selection as embodied by (17.32). If we set $\beta = 1$ and, as suggested in Chapter 15, Section 15.3, Theorem 15.3, we use α-two part MDL for $\alpha = 2$, then we are effectively picking the h minimizing

$$n\mathbf{e}_{\text{emp}}(h) + 2L(h). \tag{17.37}$$

Comparing this to (17.32), we see that the only difference is the additional term in (17.32) involving the square root of n. To illustrate the difference, suppose $\mathcal{X} = [0, 1]$, and P^* is such that $P^*(Y = 1 \mid X = x) = 1$ if $x \in [0, 0.1]$, whereas for each $h \in \mathcal{H}$, $h(x) = 0$ if $x \in [0.0.1]$. If $P^*(X \in [0, 0.1]) = 1/10$, then on approximately 10% of the sample, *all* $h \in \mathcal{H}$ will make a wrong prediction of Y_i. Then in a typical run, no $h \in \mathcal{H}$ will achieve empirical error $\mathbf{e}_{\text{emp}}(h)$ much smaller than 0.1, and then the second term in (17.32) becomes dominant: it is nonzero, and multiplied by $\sqrt{n}$. In such cases, hypothesis selection based on (17.32) will be *much* more conservative than model selection based on MDL, since in the former the weight of the "complexity" $L(h)$ of hypothesis h is multiplied by $\sqrt{n}$, and in the latter, it remains constant.

From an MDL point of view, one may now think that this additional complexity penalty implicit in PAC-Bayes is not really necessary. Indeed, the

two-part MDL consistency result Theorem 15.3 suggests that two-part MDL will be consistent. If this were the case, then it might be advantageous to drop the additional term in PAC-Bayes and use MDL instead: if all hypotheses h with smallest generalization error have large complexity $L(h)$, then PAC-Bayes will only start selecting good approximations of h for much larger sample size than two-part MDL.

Unfortunately though, Theorem 15.3 does not apply in the present situation. The reason is that the entropified model class $\mathcal{M}_{\mathcal{H}}$ will in general be severely *misspecified*: it has been artificially constructed, and there is no reason at all why it should contain the assumed true distribution P^*. Indeed, Grünwald and Langford (2007) show that the two-part MDL approach to classification (17.37) can be *inconsistent*: they give an example of a true distribution P^*, a hypothesis class $\mathcal{H}$ and a codelength function L such that there exists a $\tilde{h} \in \mathcal{H}$ with small codelength $L(\tilde{h})$ and with generalization error $\mathbf{e}(\tilde{h}) = \epsilon$ close to 0 relative to P^*; yet with P^*-probability 1, as n increases, the two-part MDL criterion (17.37) will keep selecting h with larger and larger $L(h)$, and all of these h will have generalization error $\mathbf{e}(h) \gg \epsilon$. The difference between $\mathbf{e}(\tilde{h})$ and the generalization error for all of the h selected by MDL can be as large as 0.15. One consequence of this phenomenon is that MDL as well as Bayesian inference can be inconsistent under misspecification, even with countable model classes; see (Grünwald and Langford 2007). The underlying reason for the inconsistency is, once again, the *nonmixability* of the $0/1$-loss. In the individual sequence prediction framework with finite $\mathcal{H}$, this nonmixability implied worst-case regrets of $\text{ORDER}(\sqrt{n})$. This implies that MDL based on the entropification procedure (which, if it worked, would promise worst-case regrets of constant order) cannot be applied on all sequences. In the statistical learning framework (data i.i.d. P^*, P^* unknown), the nonmixability implies that consistent hypothesis selection algorithms need $\sqrt{n}$-factors in front of the hypothesis complexities. Thus, MDL based on the entropification procedure (which would promise complexity penalties without sample-size dependent multiplicative factors) does not converge for all P^*.

Not surprisingly then, earlier approaches that try to combine learning theory and MDL-type inference for classification (Barron 1990; Yamanishi 1998) also end up with a factor in front of the hypothesis complexity $L(h)$ that can be as large as $\sqrt{n}$, and the resulting criteria have no natural coding interpretation any more; see also Meir and Merhav (1995), who do classification with one-part universal codes based on the entropification–construction (17.36).

By now, the reader may have come to wonder why we chose $\beta = 1$. From

an MDL point of view, a much more natural approach is to try to *learn* β from the data. This idea was investigated by Grunwald (1998,1999), who showed that the β learned from the data has an interesting interpretation as a 1-to-1 transformation of an unbiased estimate of the generalization error of the h selected by MDL. Adjusting (17.37) to learn β as well, (17.37) becomes: minimize, over $h \in \mathcal{H}$,

$$n\mathrm{H}(\mathbf{e}_{\mathrm{emp}}(h)) + 2L(h) + \frac{1}{2}\log n, \tag{17.38}$$

where the $(1/2)\log n$ term is used to encode β. It plays no role in the minimization and can be dropped. The value of β that is adopted is given by $\ddot{\beta} = \ln(1 - \mathbf{e}_{\mathrm{emp}}(\ddot{h})) + \ln(\mathbf{e}_{\mathrm{emp}}(\ddot{h}))$; its occurrence in (17.38) is not visible because we have rewritten $\mathbf{e}_{\mathrm{emp}}(h)$ in terms of β. Grünwald (1998) shows that several versions of MDL for classification that have been proposed in the literature (Quinlan and Rivest 1989; Rissanen 1989; Kearns, Mansour, Ng, and Ron 1997) can all be reduced to variations of (17.38). Unfortunately though, learning β from the data does not solve the serious inconsistency problem mentioned above. In fact, in their main result Grünwald and Langford (2007) show that (17.38) can be inconsistent; the inconsistency for fixed β follows as a corollary.

Summary: MDL and Learning Theory We have seen that the algorithms used in learning theory are based on the frequentist design principle, which we criticized in Section 17.1. Nevertheless, the approach is quite "agnostic," in the sense that very few assumptions are made about the underlying P^*. Therefore, it is worrying that MDL approaches to learning classifiers relative to the $0/1$-loss can fail asymptotically when investigated within the learning theory framework. The underlying reason seems to be the nonmixability of the $0/1$-loss function that we discussed in Section 17.9.

MDL and Learning Theory
In learning theory, complexity of a class of functions $\mathcal{H}$ is usually still measured in terms of quantities related to bits; in the PAC-Bayesian approach, it is directly measured in bits. But to get algorithms with guaranteed consistency, one needs to combine the complexity with the empirical loss in a more subtle way than by merely adding them.

Apart from the advantage of guaranteed consistency, the learning theory approach also has significant drawbacks compared to MDL. One problem is

that its domain of application is quite limited. For example, if the x_i are set by humans (as they often are in regression problems, viz. the term "design matrix"), then the learning theory analysis is not valid anymore, since it requires the X_i to be i.i.d. In practice, MDL and Bayesian approaches to classification often work just fine, even under misspecification. In contrast, approaches based on learning bounds such as (17.32) often need a lot more data before they produce a reasonable hypothesis than either MDL or Bayes.

> In 2002, I attended a workshop called "Generalization Bounds < 1." The title says it all: researchers at this workshop presented some of the rare cases where bounds such as those in Proposition 17.1 actually produced a nontrivial bound ($\mathbf{e}(h) < 1$) on some real-world data set. At the workshop, it turned out that in some cases, the bound was still larger than 0.5 — larger than the trivially obtained bound by randomly guessing Y using a fair coin flip!

It seems that learning theory approaches are often too pessimistic, whereas the MDL approach can sometimes be too optimistic.

17.11 The Road Ahead

Problems with MDL In this chapter we argued that from a theoretical perspective, MDL approaches compare favorably to existing approaches in several respects. In many cases, MDL methods also perform very well in practice. Some representative examples are Hansen and Yu (2000,2001), who report excellent behavior of MDL in regression contexts; the studies in (Allen, Madani, and Greiner 2003; Kontkanen, Myllymäki, Silander, and Tirri 1999; Modha and Masry 1998) demonstrate excellent behavior of prequential coding in Bayesian network model selection and regression; many more such examples could be given. Also, "objective Bayesian" model selection methods are frequently and successfully used in practice (Kass and Wasserman 1996). Since these are based on noninformative priors such as Jeffreys', they often coincide with a versions of "refined" MDL and thus indicate successful performance of MDL.

Yet there is also practical work in which MDL is not competitive with other methods (Kearns, Mansour, Ng, and Ron 1997; Clarke 2004; Pednault 2003).[16] Not surprisingly then, there are also some problems with MDL from

16. But see (Viswanathan., Wallace, Dowe, and Korb 1999) who point out that the problem of (Kearns, Mansour, Ng, and Ron 1997) disappears if a more reasonable coding scheme is used. Clarke (2004) actually considers Bayesian methods, but MDL methods would work similarly in his examples.

a theoretical perspective. These are mostly related to MDL's behavior under frequentist assumptions. A related problem is that in its current state of development, MDL lacks a proper *decision theory*. Let us discuss each of these in turn.

MDL Consistency Peculiarities In Chapter 16 we showed that the three main applications of MDL, prediction, hypothesis selection and model selection, generally have very good consistency properties: the prequential, two-part or model-selection based MDL estimator typically converges to the true distribution at near optimal rate. In Section 17.2.2 of this chapter we even saw that in nonparametric settings, consistency of predictive MDL estimators is guaranteed, even in cases where Bayesian inference can be inconsistent. Yet, as we also argued in Chapter 16, each of the three versions of MDL has its own peculiarity: for prequential MDL, we get consistency in terms of Césaro rather than ordinary KL risk; for two-part MDL, we have the $\alpha > 1$-phenomenon; and for MDL model selection, there is the curious Csiszár-Shields inconsistency result. It seems that in nonparametric cases, straightforward implementations of all three versions of MDL sometimes incur an additional $\log n$-factor compared to the risk of the minimax optimal estimation procedure. All this may not be of too much practical interest, but from a theoretical perspective, it does show that some aspects of MDL are currently not fully understood.

The problem is more serious, and presumably, much more relevant in practice, if the true distribution P^* is not in the (information closure of) the model class $\mathcal{M}$; indeed, this seems to be the main cause of the suboptimal behavior reported by (Clarke 2004; Pednault 2003). As explained in the previous section, in that case, MDL (and Bayes) may be inconsistent, no matter how many data are observed (Grünwald and Langford 2007). This is a bit ironic, since MDL was explicitly designed *not* to depend on the untenable assumption that some $P^* \in \mathcal{M}$ generates the data. Indeed, if we consider the *accumulated log loss* of the prequential MDL estimator in the inconsistency example of Grünwald and Langford (2007), we find that MDL behaves remarkably well. In fact, the problem is caused because for large n, the prequential MDL estimator $\bar{P}_{\text{Bayes}}(X_{n+1} \mid X^n)$ is a distribution on $\mathcal{X}$ that is *closer* to P^* in KL divergence than the $\tilde{P} \in \mathcal{M}$ that achieves $\min_{P \in \mathcal{M}} D(P^* \| P)$. While $\bar{P}_{\text{Bayes}}(X_{n+1} \mid X^n)$ is a better predictor than $\tilde{P}$ in terms of expected log loss (KL divergence), it is a mixture of $P \in \mathcal{M}$ each of which is extremely far from P^* in terms of KL divergence. Therefore, the posterior puts nearly all

its mass on very bad approximations of P^*, and we cannot say that $\bar{P}_{\text{Bayes}}$ is consistent. Also, if $\bar{P}_{\text{Bayes}}(X_{n+1} \mid x^n)$ is used for $0/1$-loss prediction, then it will become much *worse* than $\tilde{P}$; see (Grünwald and Langford 2007) for a thorough explanation of why this is problematic. The strange phenomenon that inconsistency is caused by $\bar{P}_{\text{Bayes}}(X_{n+1} \mid X^n)$ predicting *too well* is related to what I see as the second main problem of MDL: the lack of a proper decision theory.

Lack of MDL Decision Theory It is sometimes claimed that MDL is mostly like Bayesian inference, but with a decision theory restricted to using the logarithmic utility function.[17] This is not true: via the entropification device, it is possible to convert a large class of loss functions to the log loss, so that predicting data well with respect to log loss becomes equivalent to predicting data well with respect to the loss function of interest. Nevertheless, as we discussed in the previous section, this is not without its problems. It can only be used if the loss function is given in advance; and it can fail for some important loss functions that may be defined on the data, such as the $0/1$-loss.

More generally speaking, in Section 17.2.1 we made clear that parts of Bayesian statistical decision theory (maximize expected utility according to the posterior) are unacceptable from an MDL perspective. But this was a negative statement only: we did not give a general MDL rule of exactly how one should move from inferences based on the data (two-part MDL or prequential MDL estimators) to decisions relative to some given loss or utility function. The entropification idea gives a partial answer, but we do not know how this should be done in general. To me, it seems that what is really lacking here is a general MDL decision theory.

Conclusion Personally, I feel that the two problems mentioned above are strongly interrelated. The main challenge for the future is to modify and extend the MDL ideas in a non-ad hoc manner, in a way that avoids these problems. I am confident that this can be done — although the resulting theory may perhaps become a merger of MDL, the most agnostic brands of Bayesian statistics, prequential analysis, and some types of universal individual sequence prediction, and those statistical learning theory approaches in which complexity is measured in bits. All these alternative methods have

17. Again, I have heard people say this at several conferences.

some overlap with MDL, and they may all have something to offer that current MDL theory cannot account for. One aspect of MDL that I do not sufficiently recognize in any of the alternative approaches, is the view that models can be thought of as languages, and the consequence that *noise* relative to a model should be seen as the number of bits needed to describe the data once the model is given.

As a final note, I strongly emphasize that none of the problems mentioned above invalidates the fundamental idea behind the MDL Principle: *any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally*. The problems mentioned above suggest that this statement cannot be strengthened to "every good learning algorithm should be based on data compression." But, motivated by Theorem 15.3 and the entropification idea, I firmly believe the following, weaker statement: "every statistical estimation algorithm, every sequential prediction algorithm with respect to any given loss function, and every learning algorithm of the type considered in statistical learning theory, can be transformed into a sequential data compression algorithm. If this algorithm does not compress the given data at all, it hasn't really learned any useful properties about the data yet, and one cannot expect it to make good predictions about future data from the same source. Only when the algorithm is given more data, and when it starts to compress this data, can one expect better predictive behavior. Summarizing:

Concluding Remark on The MDL Philosophy
One cannot say: "all good learning algorithms should be based on data compression." Yet one *can* say: *if one has learned something of interest, one has implicitly also compressed the data.*